

*Your*

# COMMODORE

**YOUR BEST INDEPENDENT COMMODORE MAGAZINE**

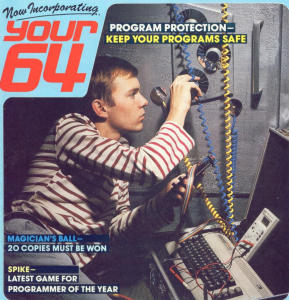
*Now Incorporating*

**YOUR  
64**

**PROGRAM PROTECTION—  
KEEP YOUR PROGRAMS SAFE**

**MAGICIAN'S BALL—  
20 COPIES MUST BE WON**

**SPIKE—  
LATEST GAME FOR  
PROGRAMMER OF THE YEAR**



# COMMODORE 64



"DRAGONSKULLS", "OUTLAWS", "BLACKWYCHE", "IMHOTEP" recommended retail price  
£9.95 inc. VAT. Available from W.B.SMITHS, BOOTS, J.PENZIES, WOOLWORTHS  
and all good software retail outlets. Also available from  
ULTIMATE PLAY THE GAME, The Green, Ashby-de-la-Zouch, Leicestershire LE6 5JU  
(P&P are included) Tel: 0530 411465

ULTIMATE  
PLAY THE GAME

FEBRUARY 1986

VOLUME 2  
NUMBER 6

Editor:  
Steven Calder

Assistant Editor:  
Marie Curry

Advertisement Manager:  
Mike Segalov

Advertisement Copy  
Control:  
Laura Champness

Group Editor:  
Gavin Bradshaw

Group Managing  
Editor:  
Wendy Palmer

Managing Director:  
Peter Neufman

Circulation:  
Scorp Typesetting

Design:  
Angus Design

Editorial & Advertisement Office  
No 1 Golden Square,  
London W1R 3AB  
Telephone: 01 467 0626  
Telex: 8871359

Your Commodore is a monthly  
magazine appearing on the first  
Friday of each month.

Distribution by: Angus Press  
Sales & Circulation Ltd, 12-18  
Paul Street, London EC2A 4JE.  
Printed by: Advertiser Presswork  
& Sales Ltd, 70-71, Mark Lane,  
Kent.

Subscription rates upon  
application to: Your  
Commodore Subscriptions  
Department, Angus Press  
House, 175-181 Mark Lane,  
London, EC3R 7HT, Marks  
Lane, 1986.

The contents of this publication  
including all articles, designs,  
plans, drawings and programs,  
and all copyright and other  
intellectual property rights  
therein belong to Angus  
Specialist Publications Limited.  
All rights are reserved by the Law  
of Copyright and other  
intellectual property rights and  
by virtue of international  
copyright conventions are  
specifically reserved to Angus  
Specialist Publications Limited  
and any reproduction without  
the prior written consent of the  
Company © 1986 Angus  
Specialist Publications Limited.

## FEATURES

■ <b>Count on Your Commodore</b> .....	7
Year 84 learns to add up.	
■ <b>Stop Thief!</b> .....	12
Get those programs under lock and key.	
■ <b>Interrupts</b> .....	32
Introducing IRQ interrupts.	
■ <b>The Bear Goes On</b> .....	43
Switch on to Syntrex's Digidrum.	
■ <b>Now Hear This</b> .....	44
Rainbird's Music System under the microscope.	
■ <b>Gremlin Grilling</b> .....	52
Inside a little monster's office.	
■ <b>Do You Need Your Head Examined?</b> .....	86
Dave Crisp reviews a utility to re-align your disk drive.	

## SERIES

■ <b>Mach 4</b> .....	18
Steve Carrie adds a Machine Code disassembler.	
■ <b>Froggy</b> .....	40
More to add to your arcade game design.	
■ <b>Language Lab - Pilot</b> .....	54
Make your 64 bilingual.	
■ <b>Build a Better Basic</b> .....	62
Your Basic grows still more.	
■ <b>Programming Projects</b> .....	70
Archaeology and microc ds mix.	

## REGULARS

■ <b>Data Statements</b> .....	4
■ <b>Scratch Pad</b> .....	10
■ <b>Teacher's Pet</b> .....	30
■ <b>In Arcadia</b> .....	36
■ <b>Game of the Month</b> .....	38
■ <b>Action Replay</b> .....	46
■ <b>Misives</b> .....	52
■ <b>Sense of Adventure</b> .....	72
■ <b>Communications Corner</b> .....	74
■ <b>Easy Entry</b> .....	84
■ <b>Listings</b> .....	87

## COMPETITIONS

■ <b>Magician's Ball Competition</b> .....	16
A chance to win Global Software's new adventure.	
■ <b>Sprite Ideas</b> .....	82
Create a sprite and win £15.	

## GAMES AND UTILITIES

■ <b>Break The Speed Limit</b> .....	22
High speed tape for your C-16 and Plus/4.	
■ <b>Spike - Programmer of the Year</b> .....	56
A great game from a great programmer.	

C  
O  
N  
T  
E  
N  
T  
S

Soft in the Head ~

YES, IT HAD TO HAPPEN SOME TIME. Superman has got fed up of doing his quick change act in the confines of a phone box and will now be executing this incredible feat of contention inside your computer. Beyond has recently launched *Superman: The Computer Game*, which features Superman, on the side of good, fighting Darkseid for control of a metropolis.

According to Beyond, the game contains a minimum of rules and is punctuated with breathtaking animated sequences. It's not a bird or a plane but it is *DSO*.

Also in Beyond's autumn release package were *Intelligence*, the sequel to *Shadowline* and *Spy vs Spy: The Island Capot*, sequel to *Spy vs Spy* (as if you hadn't guessed). Both cost £9.95 on cassette.

Another superman, international goalkeeper Ray Clemence, has got his stamp of approval on Maximilian Software's new release *World Cup Soccer*. The package contains two programs, and a book which provides information on the skills techniques and tactics of some of the world's top players. Goals Ray commented: "World Cup Soccer is a must for any serious and dedicated soccer fan. Once you've mastered this you can sit back and watch the 1990 World Cup through the eyes of a real professional."

# DATA STATEMENTS



Goalie Ray with the team

Five for under a fiver







Fred and Brent Casey Jones on computer by Microsoft

Demarc, meanwhile, has decided to try and score easily by launching *Archie the Fish, The Computer Game*, featuring a mad fossil called Jason who wanders around a holiday camp trying to get his homicidal way with all the innocent campers. Your job is, of course, to save them but mind you don't panic as this seems to inflate the geology. \$9.95 on cassette and \$11.95 on disk, it's probably a game not to be played in the dark, the radio's *Microreview*!

AtariSoft has launched a new range of software for the future which is unusual because the disk versions are under £10—at £9.95. *Planet Bringer*, marketing and sales director, said: "The case may be lower but the quality certainly isn't." The cassette versions are £7.95 and the new titles are: *Anti Assault*, *D-Box*, *Bag Office*, *Enter and Saver Attack*. All are available on the C64.



Quakeba has come up with an arcade strategy game for the 88 called *Destroba*, which features you as a brilliant artificial being to rebuild your shattered forces so that you can destroy the enemy's special research lab, hidden in the depths of a mountainside. If you don't, then you'll finish developing the *Ultimate Weapon*—an atomic bomb. (Haven't some-one already developed it?) If you want to gain lost territory from the evil Alliance then you'll have to fork out £7.95 for the privilege.

Ultimate has two new titles for the C64: *Dragon Skulls* and *Outlaw*—shouldn't that be *Autism*? They both cost £9.95 on the C64 and are embossed with Ultimate's unmistakable brand of art work.

Hitman Commodore want you to let the train take the strain and has *Southern Belle* for the C64. This steam locomotive simulator is said by its makers to be for the more 'sophisticated' game player and it is rumoured that 'railway enthusiasts have even been buying computers specifically to sample its delight.' Holy smoke!

On to more serious software, and Tempus has produced a program called *Font Factory* which is aimed at improving the output from a dot matrix printer. It reads any standard Commodore ASCII file, automatically formats and prints it. And you get a choice of eight different typefaces. It incorporates control of line width and spacing and justification. Also on the disk is a program called *Type Master* which allows you to produce banners using letters a foot high. You get both programs for £9.95.

Tempus has also released *Fantastic Filer* and *Screen Dumper 64*. Both these programs are £12.95 each and available on disk only.

## In Touch

MICRONET HAS MOVED INTO the glamorous world of pop-music. *Teegal Sharkey* appeared on *Celebrity Chatline* and was greeted with an onscreen response from Microsoft members.

Teegal has himself been a member of Microsoft for over a year and is very impressed with the service, saying: "Most of all I find it good relaxation." He also loves the *Celebrity Chatline* in particular. "I think it's very entertaining," he raved. "It's certainly a lot more entertaining than most of the programmes on TV at the moment. I'd rather watch *Chatline* than *Comedian Street*." Obviously an *Entertainment* fan.

Microsoft members have also been making an effort to help the survivors of the Mexican earthquake. In the first month of the Mexican Aid Appeal they raised over £100. Donations should be made payable to Mexican Aid and sent to the address below. Microsoft members should call page 7608119846.



1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26



Voluntary, Unilateral, Non-Reciprocal



1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26



## Headlines

MOORE ELECTRONICS HAS NOW announced the availability of a Telnet Adapter for the Cid. The Cid version of the Adapter plugs into the user expansion port and uses software to produce a console/Telnet display.

The Cameroonian system runs about 91m and anyone interested should contact it for more details.

There's also a vast range of colour options now available from Philips. There are four models in the range and prices start at £109.

Philips policy is to improve the clarity, readability and performance of monitors to bring us first with improvements made

to home computers. Philips states that the monitors are designed to give superior quality and crispness for every computing need.

## General's Southwest

**FINISHED HAS SOME GO TO A BABY SISTER** - or should that be egg. British Telecom is forming a new software company, which will be totally separate from Finland. The new outfit is to be called Rainbird and will be headed by its namesake Tony Rainbird.

First release from Rainbird in Island Logic's, The Magic System. It has been available for the BBC computers for some time, but only now has a Commodore 64 version been published.

Ramirez will also be producing software for 16-bit computers including Commodore's latest line offering, the Amiga. Farland will continue to burn brightly and independently producing games for 8-bit machines.

But those who are concerned about the laxation of the new Data Protection Act 1984, the Data protection Registrar has produced a handy questionnaire and answer booklet to try and clarify the most important points. The Act is designed to protect individuals' rights by allowing them to have access to personal information which various organisations may have on file. Subjects covered range from personal data held at schools and universities, to registration for groups of companies.

If you've fed up with seeing those horrible tangled up wires around your computer, then Comstock Electrical Limited may have come up with the solution to your problem. Now available are two new adaptors, one of which can take up to six plugs, the other up to four. Both are smaller and lighter than traditional ones, and they certainly look a lot better, too.

Both come complete with plugs and are for use in any standard 11 amp socket. They also conform to the Electrical Equipment Safety Regulations, 1980.

**Twelve Line**

Department, 2nd Floor, Leinster Court, The  
Fairlington Road, London EC1M 3AD, 07 842  
7800

Intermillenium Publishers, 4 Little Ferry St.  
Lewiston, ME 04203, U.S.A. (Tel. 603-883-6100)  
Fax: 603-883-6110

Arbuckle, 11 Westminster Palace Gardens,  
Ampleforth Place, London W87P 1B.

Qualification: Liberty Hall, 212 Ferguson St.  
Baltimore, MD 21201, tel. 410/685-6666.

Ullmann, T. H. C. *Archiv der Gesch. d. Naturh.* 1884, 20, 1-10.

Dr. William H. Anderson, Owner, 8005 132nd St.

[illegible][illegible]

**Adamsville Electric Co., 80 Mainway Pl.**

Navigation: Home | About Us | Contact Us | Privacy Policy | Terms of Service | Sitemap

Philippe, Barthelemy-Schaeffer, 24 Nov 1991  
Paris, France 94-98, 2003, 60 000 0014

British Telecom, Wellington House, Upper  
St Martin's Lane, London EC1

The Urban Postcardbook English as a Second Language  
 Edition, Wayne State, Milwaukee, WI, 1982

© 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678,

Chapman, A. de, *Woolly aethiops* n. sp.  
 holotype, NY 1001, coll. 1912.17

Intelligence Affs., 2014-2015, 1000  
 Lower, January 2015, 1000

00079792, New West Bank, Kamish Tawer  
Bamash, London 2000, 2004

In a mathematical  
special, Nick  
Hampshire shows you  
how to use the C64's  
arithmetic routines.

## Numeric Variables, Types and Range

BASIC USES TWO DIFFERENT types of numbers, integer and floating point. An integer number is stored in two bytes giving a 16-bit signed number which can store numbers in the range -32767 to +32768. Floating point numbers require five bytes and can store much larger values in the range  $\pm 1.70941183 \times 10^{-38}$  to  $\pm 2.9872988 \times 10^{38}$ . In the Basic Interpreter all calculations, whether on integer or floating point values, are performed using the latter rather than simple integers or binary values. Consequently, all integer values are first converted to floating point format before any calculations are performed.

The format for the storage of an integer value is very simple, consisting of two bytes stored in low order/high order byte. Negative values are stored in a two's complement form, — the format is shown in Figure 1. Floating point values are stored in either packed form, occupying five bytes, or unpacked form in six bytes. Packed format is the normal mode for storing floating point

variables in memory. Unpacked format is used when performing calculations upon floating point values. In either format there are three components of a floating point value, — the sign, the exponent and a four byte mantissa. In packed mode the sign is stored as bit seven of the most significant byte of the mantissa. In unpacked format the sign occupies its own byte.

## The Floating Point Accumulator

In order to perform arithmetic operations on any floating point value the interpreter needs temporary storage locations for the values being worked upon as well as the result. These are two principle work areas, they are known as

# COUNT ON YOUR COMMODORE

floating point accumulator 1 and floating point accumulator 2. These names are usually shortened to Fac 1 and Fac 2. Each floating accumulator occupies six bytes and Fac 1 starts at \$47 while Fac 2 starts at \$65. There are, in addition,

three further areas where floating point numbers in packed format (occupying five bytes) are stored. These areas start at \$57, \$5C, and \$68. The format and location of the two floating accumulators is as follows:

Locations		Function
Fac 1		
\$47	\$48	exponent + \$60
\$42	\$4A	mantissa byte 4
\$43	\$4B	mantissa byte 3
\$44	\$4C	mantissa byte 2
\$45	\$4D	mantissa byte 1
\$46	\$4E	sign (\$4F = - and \$48 = +)

```

5 REM ** REAL NUMBER FORMAT (PACKED) **
10 A=0
20 C=PEEK(45)+PEEK(46)*256+2
30 INPUT B
35 IF B=8 THEN PRINT B:GOTO 0:PRINT "GOTO 330"
40 EX=INT(C/1000000)*1000+277
50 E=C%128
60 R=B-256X
70 S0=S0*(E-8)*64+64
80 T0=T0/256X*128
90 M1=INT(T0)+S0
100 T1=(T0-INT(T0)*256+256
110 M2=INT(T1)
120 T2=(T1-INT(T1)*256+256
130 M3=INT(T2)
140 T3=(T2-INT(T2)*256+256
150 M4=INT(T3)
160 PRINTS:M1:M2:M3:M4
170 PRINT
180 FOR EC=1
190 FOR ED=1:M1
200 FOR ED=2:M2
210 FOR ED=3:M3
220 FOR ED=4:M4
230 PRINTM

```

Program 1

```

5 REM ** REAL NUMBER FORMAT (PACKED) **
10 A=0
20 C=PEEK(45)+PEEK(46)*256+2
30 INPUT B
35 IF B=8 THEN PRINT B:PRINT "GOTO 330"
40 EX=INT(C/1000000)*1000+277
50 E=C%128
60 R=B-256X
70 S0=S0*(E-8)*64+64
80 T0=T0/256X*128
90 M1=INT(T0)+S0
100 T1=(T0-INT(T0)*256+256
110 M2=INT(T1)
120 T2=(T1-INT(T1)*256+256
130 M3=INT(T2)
140 T3=(T2-INT(T2)*256+256
150 M4=INT(T3)
160 PRINTS:M1:M2:M3:M4
170 PRINT
180 FOR EC=1:M1
190 FOR EC=2:M2
200 FOR EC=3:M3
210 FOR EC=4:M4
220 PRINTM

```

Program 2

other locations used are:  
 \$ld = overflow byte for fac 1  
 \$ld—sign comparison byte  
 \$ld—overflow byte for fac 2

### How a Floating Point Number is Stored

The storage of a floating point number is fairly complex both in packed and unpacked format. The data need to meet a floating point number can be divided into three components: the exponent, the sign and the mantissa. In the unpacked format, the exponent and sign each occupy one byte and the mantissa four bytes. The following is an explanation of each component of a floating point number.

**Exponent** — The exponent indicates the position of the decimal point within the number. The seven of the exponent byte indicates the sign of the exponent. That is, if the exponent is positive, bit seven is set to one and, therefore, the value of the exponent byte will always be greater than 128. If the exponent is negative then bit seven is set to zero and the exponent value is less than 128. The exponent is stored as a power of two and is multiplied by the mantissa value to produce the final value. The following formulae can be used to convert a number *N* stored in the mantissa bytes (see paragraph on mantissa for calculation of *N*) into the floating point number by multiplying it with a positive

Volume 14 Number 2 2004

To determine the exponent of a number, find the highest power of two which can be subtracted from the number. Thus, if the number is 18,254, then the highest power of two is 16,384. The remaining value is positive, and therefore equals 18,254 or 1.71. The fact that the exponent is divided in this way means that the mantissa for two different values may be the same, although the difference being registered solely by the exponent of the exponent. Thus, the floating point mantissa controls for the values 1.19179 [40] and 0.26125 [41] but is unaltered.

1.14710 stored as — equivalent  
 110 and modulo 21,79,270,161  
 6.26318 stored as — equivalent  
 110 and modulo 21,79,270,161

As you can see, multiplying and dividing a floating point number by two is a very simple operation involving adding or subtracting one from the exponent. The range of the exponent is  $\pm 2778$ . This equates approximately to  $10^{\pm 845}$ .

**Sign** — The sign of the value is stored in unpacked — format in a single byte with a value of 144 for negative numbers, for 160 for positive numbers. In packed format the sign is stored in bit seven of the highest byte of the mantissa. If bit seven is zero then the mantissa is positive, and if it is one then the mantissa is negative. Thus the unpacked floating-point values for 2.0 and

3. **IMPACT**

numbers of 10 — reported TAC and numbers left fish

numbers of 16. — exposure 150  
and moisture 150, 151

**NUMBERS** — The numbers is stored in four bytes less the most significant bit of the most significant byte of the numbers which is used to store the sign bit. The number is then stored in the numbers less its number

```

0000      | CALCULATE CH+25/CH#3
0000      | WHERE A AND B ARE INPUT FROM
0000      | THE KEYS#R#3.
0000      | ENTRY AT SVS 49071.
0000      |
0000      | RESULT IS PRINTED
0000      |
0000      |---#C#000
0000      |WV      WVR 0
0000      |WV      WVR 0
0004      |00000000 TP1    BYT 0.0.0.0.0
0008      |00000000 TP2    BYT 0.0.0.0.0
000C      |00000000 TP3    BYT 0.0.0.0.0
0010      |0000      ENTRY  LBY #000
0014      |2007FF L1      JSR #FFDF      INPUT BYTE
0018      |0000      CMP #000      CHARACTER RETURN?
001C      |F000      BDC L3      YES
0020      |00000002 STA #0000.Y    STORE BYTE
0024      |00      INY      DO NEXT
0028      |00F3      BNE L1      ALWAYS
002C      |0000      LDR #000      ZERO TERMINATOR
0030      |0000      STA #0000.Y
0034      |0000      LDR #000
0038      |0070      STA #70
003C      |0002      LDR #002
0040      |0070      STA #70
0044      |207000      JSR #0070
0048      |200000      JSR #0000      ICONVERT TO # 0-65535
004C      |20F707      JSR #0077      MAKE INTEGER
0050      |0014      LDR #14      STORE VALUE
0054      |000000      STR #V      IN TEMP
0058      |0015      LDR #15
005C      |000100      STR #V+1
0060      |0000      LBY #000
0064      |2007FF L2      JSR #FFDF      INPUT BYTE
0068      |0000      CMP #000      CHARACTER RETURN?
006C      |F000      BDC L4      YES
0070      |00000002 STA #0000.Y    STORE BYTE
0074      |00      BNY      DO NEXT
0078      |00F3      BNE L3      ALWAYS
007C      |0000      LDR #000      ZERO TERMINATOR
0080      |00000002 STA #0000.Y
0084      |0000      LDR #000
0088      |0070      STA #70
008C      |0002      LDR #002
0090      |0070      STA #70
0094      |207000      JSR #0070
0098      |200000      JSR #0000      ICONVERT TO # 0-65535
009C      |20F707      JSR #0077      MAKE INTEGER
00A0      |0014      LDR #14      STORE VALUE
00A4      |000100      STR #V      IN TEMP

```

100

equivalent use the following formulae:

$$N = (M1 \text{ AND } 127) * (M2 * M3 + M4 * 256 / 255 / 255 / 255)$$

where M1, M2, M3 and M4 are the mantissa bytes, with M1 the highest and M4 the lowest. When N has been obtained it should be multiplied by 2 (programmer — 128) to give the actual value. The program in Program 1 allows the input of a number, then prints the contents of the exponent and mantissa bytes for that number as it is stored in floating point. These values are then used by lines 90 to 126 to convert the floating point byte values back into the number.

To convert a number into floating point form is a slightly harder calculation and involves the following steps:  
First find the highest power of two which can be subtracted from the number, i.e. the value of two to its highest power.  
Secondly let R = the remainder after subtracting the value of 2 i.e.

The calculation is then as follows:

$$T0 = (R/5)*128$$

M1 = INT(T0)+mantissa sign  
(sign = 0 if positive 128 if negative)

$$T1 = (T0 - INT(T0)*5)*256$$

$$M2 = INT(T1)$$

$$T2 = (T1 - INT(T1)*5)*256$$

$$M3 = INT(T2)$$

$$T3 = (T2 - INT(T2)*5)*256$$

$$M4 = INT(T3)$$

Where M1, M2, M3, M4 are the four mantissa byte values, M1 being the highest. The program in Program 2 does this conversion of a number input at the beginning of the program into the four bytes of a floating point format which are displayed on the screen. The program then checks by putting these values into the first variable in memory defined as a simple variable A in line 10.

The following are examples of the storage of some floating point numbers:

Number	Exponent	M1	M2	M3	M4	Sign
1	501	000	000	000	000	000
-1	501	000	000	000	000	011
.5	500	000	000	000	000	000
.25	500	000	000	000	000	000
0.128	500	000	000	000	000	000
0.064	500	000	000	000	000	000

0000	0015	LBR	#15		
0001	0100C0	STR	00+1		
0001	0100C0	LBR	00+1		10ET FIRST VALUE
0004	0000C0	LBR	00		
0007	200100	JSR	00001		IFLOAT IT
0009	0004	LDR	#CTF1		1STORE IN TEMP FAC1
0010	0000	LBR	00101		
0010	200400	JSR	00004		
0011	0000	LBR	0000		1VALUE 22 (0100)
0010	0010	LBR	0010		
0015	200100	JSR	00001		IFLOAT IT
0016	0000	LDR	#CTF1		1POINT TO TEMP
0016	0000	LDR	#CTF1		1FAC1
0017	200700	JSR	00007		1ADD
0018	0004	LDR	#CTF1		1STORE IN TEMP FAC1
0011	0000	LDR	#CTF1		
0013	200400	JSR	00004		
0016	0000C0	LBR	00+1		1GET SECOND VALUE
0019	0000C0	LBR	00		
0020	200100	JSR	00001		IFLOAT IT
0021	0004	LDR	#CTF2		1STORE IN TEMP FAC2
0022	0000	LBR	#CTF2		
0023	200400	JSR	00004		
0024	0000	LBR	0000		1GET VALUE 3
0025	0000	LBR	0000		
0026	200100	JSR	00001		IFLOAT IT
0027	0000	LDR	#CTF2		1POINT TO TEMP
0028	0000	LDR	#CTF2		1FAC2
0029	200200	JSR	00002		1MULTIPLY
0030	0004	LDR	#CTF1		1POINT TO TEMP
0031	0000	LDR	#CTF1		1FAC1
0032	2000F0	JSR	0000F		1DIVIDE
0033	0000	LDR	#CTF3		1STORE RESULT IN
0034	0000	LDR	#CTF3		1TEMP FAC3
0035	200400	JSR	00004		
0036	200000	JSR	00000		1CONVERT TO STRING
0037	201000	JSR	0000E		1PRINT STRING
0038	00F400	JMP	00474		1"READY..."

Table 1

## Using the Arithmetic Routines in a Machine Code Program

Using the arithmetic routines within the Basic interpreter can save the programmer a lot of time in program development. It can also considerably reduce the size of a machine code program. The only penalty is that in the program using eight or 16 bit floating point arithmetic routines will have a considerably slower run time than specially written routines. When faced with the necessity of having to use arithmetic

routines the best procedure is to always use the interpreter routines and only replace these if the program is running too slow. A list of the main arithmetic routines within the C04 is shown in Table 1.

It is quite simple to utilize the interpreter arithmetic routines within a machine code program. The essential thing to remember is that the interpreter does all its calculations on floating point numbers, therefore all integer values must first be converted to floating point. The following is an example of a routine using the interpreter arithmetic routines:

$$\text{calculation } C = (A * 25) / (B * 5)$$

Where values A and B are both positive assigned 16-bit integer values these are both input from the keyboard at the start of the beginning of the routine

and the result C is a five byte floating point value which is both stored in memory and displayed on the screen. Variable storage locations in memory used by this routine are:

0000 — 16-bit value A  
0001 — 16-bit value A  
0002 — 16-bit value B  
0003 — 16-bit value B  
0004 to 0006 — temporary floating point value storage 1  
0007 to 0009 — temporary floating point value storage 2  
000E to 0010 — floating point result C storage

This article is extracted from the following books and readers are recommended to consult them for further information — Advanced Commodore 64 Basic Revealed and Commodore 64 ROMs Revealed both by Nick Humphreys and published by Caltex.

# Scratchpad

This month K Frost provides a couple of very handy routines for use on all machines.

HOW OFTEN HAVE YOU wanted to put a scrolling message across your screen? You know the type, they are used in most games programs to give instructions or a witty message. The first routine does just this. It is written in Basic but nevertheless is quite fast and would be very easy to include in your own programs as a sub-routine.

All the routine requires is that the message to be scrolled is held in the string A\$, and the positioning of the string is held in D\$, ie. D\$ should hold a home and a number of cursor movements.

The 10 in the A\$D\$ statement is the width of the message window. This can be any size but don't forget if you go over 40 the message will wrap over more than one line on the screen.

## All Things Bright

The second routine is one that will display a message and flash the letters of that message in different colours. This is very good for messages such as "PRESS ANY KEY TO CONTINUE" or "SPACE TO PLAY".

Again the program is in Basic and you can easily add it as a sub-routine to your own programs. The message that you wish to colour should be held in the string A\$. Should all the colours through which you wish the letters to cycle. Do experiment with this as some very interesting effects can be achieved.

### PROGRAM: COLOUR / K.FROST

```
100 PRINT"CLEAN":REM COLOURS
    K.FROST 1985
110 POKE 53280,0:POKE 53281,0
120 REM A$ IS THE STRING TO DISPLAY
130 A$="YOUR COMMODORE"
    :BB="WHITE,RED,CYAN,MAGENTA,
    GREEN,BLUE,YELLOW,CB,CB,CB,CB,
    CB,CB,CB"
140 REM * MAIN ROUTINE *
150 FOR A=0 TO 15:PRINT"HOME,
    DOWN,RIGHT";:REM POSITION
    THE STRING
160 FOR B=1 TO LEN(A$)
    :C=INT(RND(1)*4)
    :PRINT MID$(BB,C,1):MID$(A$,B,
    1);
170 FOR X=0 TO 15:NEXT X,B,A
```

### PROGRAM: SCROLL / K.FROST

```
100 PRINT"CLEAN":REM SCROLLING
    K.FROST 1985
110 REM A$ IS THE STRING THAT
    YOU WANT TO SCROLL
120 A$="THIS IS AN EXAMPLE OF
    SCROLLING FOR YOUR COMMODORE"
130 REM D$ IS USED TO POSITION
    THE MESSAGE ON THE SCREEN
140 REM CHANGE THIS TO SUIT YOUR
    OWN NEEDS
150 D$="HOME,DOWN,RIGHT";
160 REM * MAIN ROUTINE *
170 BB=CHR$(20):CB=BB+BB+BB+BB
180 A$=CB+CB+CB+CB+CB+CB+CB+CB
    :FOR A=1 TO LEN(A$)
190 REM THE 10 IN THE NEXT LINE
    IS THE WIDTH OF THE MESSAGE
    WINDOW.
200 REM CHANGE THIS TO SUIT YOUR
    OWN NEEDS.
210 PRINT D$:MID$(A$,A,10);
    CHR$(145)
220 FOR X=0 TO 60:NEXT X,A
```

# B.B.B. BOUND TO BE A HIT.

"10 levels of fun make this torture excellent value for money. News rating. Definitely one of those 'just another go' games. Game of the month February" - Computer Gamer

"The most compulsive game I've ever played. If you don't buy it you'll never know what you've missed." says Gary Penn, Zzap! 64, Gold Medal Award, 90% overall.

**£9.95**

cassette

**£12.95**

disc

Free with Boulder Metabolis.  
Is it a man, is it a bird?  
(Commodore 64/128 version only)



# B.B.B. BOUNDER...

Combs Graphics Software Limited, Alpha House, 1st Corner Street, Salford M6 4PL, Tel: 0161 7534424

**CBM  
64/128**

Available  
Revised late  
January



**Eric Doyle shows you  
how to foil the  
program pinchers.**

# SITTING ON THE THIEF

AFTER SPENDING MANY A sleepless night and countless day creating and debugging your latest computer masterpiece, it's disconcerting to know that any Tom, Dick or Harriette can rip it off in seconds flat. In the past many methods have been suggested to prevent LISTING but few are satisfactorily secure.

The four most common methods prevent the list function from operating properly but only one of these works after the program has been RUN.

Firstly, there is the simple expedient of using a shifted 'L' in a BASIC statement or the first line of your program:

```
10 REM [L]
```

Trying to list a printer causes it to hang up after the REM and the normal command LIST merely produces the following:

```
10 REM
SYNTAX ERROR?
READY
```

To undo this protection it is merely a case of deleting line 10 and then the program can be freely listed by anyone. Not very secure.

The second method only protects one line of a listing and also uses the protection of the REM statement. The easiest way to use this operation is to enter a line such as

```
10 PRINT"HELLO" REM
```

Press the return key to enter the line and then move the cursor to the space after the quotes. Press the shifted delete key (DEL) four times and then press it again four times unshifted (END). This should give four reversed letter 'I' symbols. Next type GOTO 10 and press return.

If you now LIST the one-line program you should see

```
10 PRINT"HELLO" GOTO 10
```

As there is no line 10 an error message would be

expected but when this program is RUN it executes normally with no error. All that has happened is that the REM part of the line has been masked by the delete symbols and the GOTO is still seen by the operating system as being within a REM statement. When the system tries to LIST to the screen the deletes are executed and this effectively pulls back the GOTO over the REM. On a printer the trick is shown in its original form as a reversed I.

Instead of the intricate INVT DEL routine, just pressing RVS ON and the letter I gives the same effect but with less fiddling about.

Adding more deletes pulls the GOTO further back along the line and experimentation will show that the command can be pulled back over the line number and even on to the previous line.

This is useful because it can be used in conjunction with the first list protection method to disguise its presence.

```
10 PRINT"HELLO"
11 REM"14" delinv100
REM"15" [L]
```

This apparently only lists a line 100 on the screen and gives a SYNTAX ERROR message. The fake line 100 could succeed in getting most people off the screen and using line numbers which are not divisible by 10 would make deletion of the coded lines difficult. On a printer the trick is revealed as the deletes are shown in their original form as reversed I's.

Instead of the PRINT statements in line 10 you could use a PRINT statement which looks for the shifted L in line 11.

To find the location of this character insert the following line:

```
10 IF PEEK(XXXX) <> 204
THEN SYS 6478
```

Now enter the following in direct mode (no program line number)

```
FOR A=2049 TO 5000:IF
PEEK(A) < 204 THEN
NEXT
```

When the cursor reappears type PRINT A and monitor line 10 with the number obtained (2048) in place of (XXXX). Repeat this line somewhere deep in the rest of your listing and hide it using the next method of protection. Remember that whenever line number 1 is used for the two lines at the beginning of the program, the position of the shifted L will not move.

The third method takes advantage of the way a line of Basic is seen by the operating system. A line consists of two bytes which give the memory address of the start of the next line, two bytes giving the current line number and then the tokenised code for the Basic instructions, followed by a null (zero) byte denoting the end of the line. When a line is listed the system is used to tell the system to start a new line, not the two byte pointer to the beginning of the next line. We can fool the system into jumping to the next line during a list by inserting a null byte where a list expects it causing the list to prematurely jump to the next line without listing the Basic code in the current line.

After writing your program, decide which line you want to hide and place any five letters

at the beginning of the line.

```
10 SSSSPPRINT"HELLO"
```

Next insert a GOTO at the end of the previous line or insert a new line which contains solely a GOTO command:

```
1 SSSSPP
10 SSSSPPRINT"HELLO"
```

RUN the program until the stop is reached and the familiar break message is displayed. At this point the system has stored the memory location of the next line just in case GOTO is used. Locations 41 (SSS) and 42 (PP) contain these pointers to the start of the line is given by the formula  $PEEK1 PEEK (SS) * 256 + PEEK (PP)$ . In the example the value would be 2056.

To avoid upsetting the line link and the line number add five to this value and place the location given with zero (PEEK 2061,0). The dummy GOTO command can then be deleted and a LIST will show only the line number of the hidden line whether listed to the screen or a printer.

For the final method of list protection we need to know a little about the way in which the 64's memory is organised. Locations 776 (XXXX) to 779 (XXXX) mainly consist of jump vectors for some of the main BASIC routines like LOAD, SAVE, PRINT, and, more importantly, LIST. A vector is a two byte number which gives the location of the start of the in-built machine code routine which performs the relevant task. For example, a memory map gives the LIST vector as being in locations 776 and 777 (XXXX-XXXX). Change either of the values found in these locations and the LIST function will be disabled causing all



## PROGRAM: BASIC LOADER

```

5 PRINT"PLEASE DOWNLOADING PLEASE WAIT:"FOR A=0 TO 50
10 FOR B=0 TO 7:READ C:G=0:G=POKE 4096+B,C:PRINT" ",
NEXT
20 READ C:IF C=0 THEN PRINT"DOWNLOADED"
30 GOTO 1:END"THIS IS OVER"END
40 NEXT:PRINT"PRINT"DOWNLOADED" IF A=0 TO 100:NEXT
50 INPUT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
60 IF B$="Y" THEN B=1:GOTO 70
70 IF B$="N" THEN B=0:GOTO 100
80 IF B$="?" THEN GOTO 60:PRINT "INVALID"
90 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
100 GOTO 100:PRINT"INVALID"
110 IF B=1 THEN GOTO 120
120 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
130 IF B=1 THEN GOTO 140
140 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
150 IF B=1 THEN GOTO 160
160 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
170 IF B=1 THEN GOTO 180
180 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
190 IF B=1 THEN GOTO 200
200 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
210 IF B=1 THEN GOTO 220
220 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
230 IF B=1 THEN GOTO 240
240 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
250 IF B=1 THEN GOTO 260
260 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
270 IF B=1 THEN GOTO 280
280 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
290 IF B=1 THEN GOTO 300
300 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
310 IF B=1 THEN GOTO 320
320 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
330 IF B=1 THEN GOTO 340
340 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
350 IF B=1 THEN GOTO 360
360 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
370 IF B=1 THEN GOTO 380
380 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
390 IF B=1 THEN GOTO 400
400 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
410 IF B=1 THEN GOTO 420
420 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
430 IF B=1 THEN GOTO 440
440 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
450 IF B=1 THEN GOTO 460
460 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
470 IF B=1 THEN GOTO 480
480 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
490 IF B=1 THEN GOTO 500
500 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
510 IF B=1 THEN GOTO 520
520 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
530 IF B=1 THEN GOTO 540
540 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
550 IF B=1 THEN GOTO 560
560 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
570 IF B=1 THEN GOTO 580
580 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
590 IF B=1 THEN GOTO 600
600 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
610 IF B=1 THEN GOTO 620
620 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
630 IF B=1 THEN GOTO 640
640 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
650 IF B=1 THEN GOTO 660
660 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
670 IF B=1 THEN GOTO 680
680 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
690 IF B=1 THEN GOTO 700
700 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
710 IF B=1 THEN GOTO 720
720 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
730 IF B=1 THEN GOTO 740
740 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
750 IF B=1 THEN GOTO 760
760 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
770 IF B=1 THEN GOTO 780
780 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
790 IF B=1 THEN GOTO 800
800 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
810 IF B=1 THEN GOTO 820
820 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
830 IF B=1 THEN GOTO 840
840 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
850 IF B=1 THEN GOTO 860
860 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
870 IF B=1 THEN GOTO 880
880 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
890 IF B=1 THEN GOTO 900
900 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
910 IF B=1 THEN GOTO 920
920 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
930 IF B=1 THEN GOTO 940
940 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
950 IF B=1 THEN GOTO 960
960 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
970 IF B=1 THEN GOTO 980
980 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$
990 IF B=1 THEN GOTO 1000
1000 PRINT"DO YOU WANT TO SAVE TO FILE OR NOT? (Y/N)";B$

```

manner of things to occur if the command LIST is used. Inserting a line at the beginning of your program such as:

```
IF POKELIST,POKE,POKE,POKE
```

would result in a system error if LIST was attempted. Changing those values to 121 and 124 respectively would apparently cause nothing to happen.

One word of caution, I'm all in favor of experimentation but be warned: never play around with vectors if you have something valuable in the memory, you'll lose it!

All these systems have a common fault: they are all easily circumvented once loaded. What is needed is a program which will run automatically and again we need to look at the vectors in the light of how the operating system loads the LOADS command.

A call is made to the vector jump at location 770 (\$A02) which causes a warm reset of the system. If this vector is changed to cause the newly loaded program to run we will have achieved our aim. But how can this be done?

First we must check that the necessary locations, which define the start and end of the program are correctly placed and that the next pointer is set to the beginning of the program. Fortunately, this can be done by calling up a routine in the Basic ROM located at \$1000 (\$A000) and then a call to \$1000 (\$A000) will cause the execution of the program. This means that we have to place a short routine into the memory and point the warm start vector to the start of the routine.

Finding a place to store the warm start routine can be fraught with problems because it must stay in memory to keep the program re-executing.

The cassette buffer must be avoided in case the program needs to access the recorder for any reason, but below this buffer are eight consecutive bytes of free memory which will house our six byte routine comfortably.

```

ROM ($A000)
ROM ($A000)

```

Two bytes of the warm start vector can now point to this routine and because the warm start is called up by the

ROM/STOP-BISTORE routine the use of the ROM/STOP key will result in the program restarting from the beginning.

To ensure that the warm start will work, a wedge, called from the CHECKOUT vectors at \$05(\$05), must be placed into the cassette buffer to allow the vector at \$770(\$A02) at the end of loading. When the system tries to print READY, it jumps into the wedge which changes the warm start vectors. All of the memory from \$770 to the end of your Basic program is saved after the CHECKOUT vectors have been changed.

Obviously, you cannot change the vector without using a machine code SAVE routine. The one included here starts off with a Basic program which stores the details of the load and save devices and filenames. This jumps into a machine code routine which loads the program which you want to protect, deletes a suitable loading screen and places the warm start details in memory. A save is then performed and a cold reset is performed allowing you to try your new auto-loading program out.

The Basic loader program includes a save routine. To set up your Stop Third Master enter and save the loader program on a spare tape or disk. Type in and save the Stop Third Master program and then load and run the Basic loader (or that it stores the program immediately after the Master program if you are using tape).

When using Stop Third, load the Master program and run it. This automatically loads the machine code program. Although you will receive a prompt, make sure that the tape/disk containing the program ready for conversion is in the recorder/disk.

After loading, suitable prompts will appear to enable you to successfully save your warm start program and disk users should not be too concerned about the fact that the saving screen problems that it is loading the program. Remember that the screen will be saved along with your program and therefore will become your loading screen.

The results of your labours will be unbeatable Basic programs...well almost!



# A Flash Of Genius!

## THE NEW 64 MULTIMODEM

**GIVES YOU DATABASES, BULLETIN  
BOARDS, ELECTRONIC MAIL,  
PRESTEL — ALL ON YOUR  
COMMODORE 64 OR 128**



At last! The perfect modem for your Commodore® 64 or 128, giving you access to Prestel®, Micronet®, viewdata and a host of other services — plus user-user communications. It couldn't be simpler. The 64 Multimodem has automatic and autoanswer, with all current

software on-board in ROM. 64 Multimodem fits your Commodore's cartridge port, and has just one external connection — the telephone lead.

The 64 Multimodem is menu driven and multispeed, with CDTT V31/23 and Bell 103 standards, handling baud rates of 300/300, 1200/75 and 75/1200. Functions include save and print frame and automailbox with edit and save.

Buy your 64 Multimodem now and we'll give you free introductory subscriptions to Micronet and Microlink — an unbeatable package of an unbeatable price — it's pure genius!

**£98.50**  
(exc.)



Approval applied for

Very limited time offer (supply and demand)

From the Communications  
Powerhouse



# MIRACLE TECHNOLOGY

MIRACLE TECHNOLOGY (UK) LTD. ST PETERS STREET IPSWICH IP1 1XB ENGLAND  
(0473) 216141 4 LINES TELECOM GOLD 35 KEY D01 (Diallink 72 076 10135)  
9460280 CREDIT G 18000285 PRESTEL MAILBOX 915992285

send to:  
Commodore Support, Miracle  
Technology (UK) Ltd.,  
St Peters Street, Ipswich IP1 1XB.

please flash me ☐  
64 Multimodem/s  
@ £116.15 (inc VAT & UK delivery)

I enclose cheque/postal order  
please tick my Accredited

card no. \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

Postcode \_\_\_\_\_



## COMPETITION

Make magic with this month's prizes from Global Software.

THERE'S MAGIC IN THE AIR, AND embedded too in Schlegel's Ball from Global Industries. If you want to improve your spelling, then study the illustrations on this page and see if you can be one of the lucky readers to enter an enchanted, adventuresome world. If you want to find out more about the game that turns to fiction of Adventure for details, there are 25 copies of the game to be won and the top five entrants will also receive a copy of *Schlegel's Executive magazine*.

## How to Use

STUDY THE ILLUSTRATION ON THIS page. Look very carefully at them and several unbalanced balls hidden in the pictures. When you think you have found all the hidden balls, circle them, clearly.

Put in the entry coupon carefully labeled and in an envelope, noting the number of teeth you found on the bait.

Special joint entry to: Global Software Competition, Your Communications, Pm 1 Golden Sq, London W1R 3AB. The closing date is Friday 20 February 1988.

They may enter at many times as you wish but each entry must be on an official coupon and sealed in a separate envelope.

**Important:** Please follow closely the guidelines on entering: – incomplete coupons and entries with no numbers on the back cannot be considered.

1000

ENTRIES WILL NOT BE ACCEPTED from employees of Argus Specialized Publications, Alabaster Personnel and more and Global Software. The restriction also applies to employees' families and spouses of the companies.

The House to Future section forms part of the index. The editor's decision is final and no correspondence will be returned into.



**Global Software Competitors: India**

100

© 2006 Pearson Education, Inc. All rights reserved. Printed in the United States of America. This publication is protected by copyright. Any unauthorized reproduction or distribution of this work without written permission from Pearson Education, Inc., may result in legal action.

Downloaded from <http://ajphaphysoc.org/> on November 10, 2015

### Neuroendocrine and Autonomic Nervous System Dysregulation

Send to: Clinical Software Competitions, Your Correspondence, No 1 Golden Sq, London W1R 4AB. Write clearly and don't forget to write your answer on the back of your envelope.

THE COUNTDOWN HAS BEGUN

# CRITICAL MASS

From DURELL



RLEP 6875

Commodore 64

Spectrum

**DURELL** sales dept.,

Castle Lodge, Castle Green, Taunton TA14AB

Write now for details of Durell's  
Great New Competition, also  
T-shirts, Catalogue  
and Posters

# MACH 4

## Steve Carter adds a machine disassembler to the Mach I monitor.

IN THIS ARTICLE, I'LL GIVE listings of the MACH I monitor extension and also some information on the whole MACH series of programs.

The extension adds a disassembler to the normal monitor commands. When the Basic program is RUN for the first time, it will patch the disassembler into the normal monitor code. Note that it replaces the I command (page restart), so you will not be able to use the warm start facility (a similar effect can be achieved by issuing a G 0000 command).

When you have typed in and saved the extension listing, place a disk containing the original monitor program in the drive then RUN the program.

First, the extension code is placed in memory at address 7000 hex. Each time the checksum, if a data error occurs, the program will print the number of the line where it was detected. This checksum isn't 100% reliable since one error may cancel out another so be careful!

Next, the original monitor code is located in at its usual address of 0200 hex. A series of FORB's patch the extension into the main code. These changes are as follows:

- 1 Change the I command to D and reset execution vector.
- 2 A section of code in the original monitor which sets the top of memory is altered to sit at address 7000 hex

3 Alter start-up message vector. Finally, the whole program is saved to disk under the name of NEWMACH. When you can save that everything works OK, you can replace the original MONITOR file with this new version.

When the BASIC is complete, type S/S 4070 to reset the machine (plus) switch the machine off then enter the monitor with S/S 11200 (unchanged). In addition to the usual startup message, you should see another one similar to:

MACH I EXTENSION  
VERSION 1.5  
(C) OCTOBER 1985 S.D.C.(R.C.)

Now, if you type D 000 < return > the disassembler should print out the contents of one of the ROM routines. Note that this command automatically sets hex I/O mode.

You should note that there is now about 16 less source code space for the editor which still leaves about 10K.

### The Monitor Jump Table

When I wrote the MACH series, I decided to put some of the more commonly used routines into the monitor. A jump table was provided at address 0200 hex to access these routines. Since the monitor is always present, a program only had to know where to call the required jump. A fair chunk of memory was saved using this method. In fact, looking back on it now, a lot more could have been saved.

The names of the routines and their call addresses are

given below.

Name	Address
START	0200
SAVE	0201
LOAD	0206
NAME	0209
OUTADD	020C
OUTBIT	020F
RET	0212
EVNL	0215
CHRGES	0218
CHRONUM	021B
CHRCANL	021E
INBT	0221
OPEN	0224

Function	
WARM START	
SAVE TO DISK	
LOAD FROM DISK	
FILE NUMBER	
OUTPUT A 3-BYTE NUMBER	
OUTPUT A 1-BYTE NUMBER	
OUTPUT A CHAR	
TRANSIT EXTENSION	
CHECK ACC. FOR HEX CODE	
CHECK ACC. FOR NUMBER	
CHECK ACC. FOR ALPHA	
SET FILENAME PARAMETERS	
OPEN FOR I/O CHANNEL	

A JIR to the appropriate address should be used since each routine ends with RTS (except START).

Some are more useful than others. A more detailed description follows. Note that TXPTR is the CHRGES test pointer at address 7A hex and TIR is the terminal input buffer at address 0200 hex.

START — This simply vectors to the code for entering the monitor. It is a non-returning routine which restores the 600 system stack. This is normally used by an external command to return control to the monitor (see INTERNAL COMMANDS). This is also the address 0200 used by the Basic start-up call.

SAVE — Save memory to disk. At entry, TXPTR must point to an ASCII string in TIR which has the format:

Filename < address1 > < address2 >

This is the same format as the monitor S command. Any error causes a jump to the error handler. On exit, TXPTR points to the end of the string. This routine uses a logical file number of 1. Note that the Basic ROM is switched out to allow the area 0000-00FF hex to be saved too.

LOAD — Load from disk. On entry, TXPTR points to an ASCII string in TIR with format:

<Filename>

This is the same as the monitor L command. Any error causes a jump to the error handler routine. On exit, TXPTR = end of string. Use logical file 1.

NAME — Name filename. On entry, TXPTR points to an ASCII string in TIR which represents the filename. At exit, register is as follows:

X = length of filename string  
Y = start offset of string in TIR  
TXPTR = end of string + 1

This routine uses spaces as delimiters. An error is given if string exceeds 50 characters.

OUTADD — Output a 3-byte ASCII hex as decimal string. On entry, X & Y contain the 30-bit value to be printed. The output mode (hex or decimal) depends on the flag OUTMOD (address 0240 hex). If OUTMOD is zero, output mode is decimal otherwise it is hex.

OUTBIT — Similar to OUTADD except an 8-bit value

in the accumulator is used.

**OUT** — Simply outputs a 0xH combination to the current output device.

**EVAL** — Evaluation expression. On entry TEXITR points to the start of the string. The flag OUTMODE operates in the usual way. Input will be ignored if either the number is out of range (0-0x7F0510) or illegal characters are found.

Note that this routine also checks for the apostrophe (') which puts EVAL into ASCII mode regardless of OUTMODE. On exit, TEXITR points to the end of the string +1.

**CHRG, CHRNUL, CHRNUM** — These three routines check the accumulator for an ASCII hex, numeric or alpha numeric character respectively. At exit, carry-flag set indicates a valid character.

**NAME** — Set filename parameters. On entry TEXITR points to the start of a filename in TAB. On exit, INTERNAL routine SETNAME (FIFO) has

called and TEXITR = end of filename +1.

**CODE** — Opens disk device if error channel 15. No input parameters. There is no CCLINK call. A file close may be accomplished using the following routine.

LDAA #15 ; CODE LINKS LN TAB  
(AR: 0xFFC) ; INTERNAL CLOSE  
BCOUTIME

Put simply, an external command is one whose code is not resident in the MACH system. Internals normally occupy the area from 9000 to 91FF hex but may extend up to 91FF hex if the Basic ROM is switched on (if you use this method, remember to switch the ROM back in before calling START to return to the monitor).

When the external has finished executing, a BAF START allows the monitor to regain control. The actual call address of an external is 9000 hex.

The programs MACRO and

ASSEMBLER are externals. Often, if the external you want to use has already been called it must have been the last external called, a G ROM command will provide a quicker method of execution. This is because an external remains resident after execution i.e. it is not deleted from memory. This is true until another external is called or the F command is used to clear out this section of memory.

## The Macro Processor Bug

And now, the bug! I discovered this while using the MACH system. It will only affect you if you use the macro processor on large source files containing macro calls. Basically, when the macro processor is constructing the output file, it uses the available memory from 8000 to 90FF hex. Due to a programming oversight (a nice way of saying I made a mess) no check is made

to see if this file is overflowing into the area above 7000 hex. The processor will quite happily destroy itself!

If the application you are working on needs no macros then you can leave out the macro processing altogether and submit the raw code to the assembler.

Unfortunately, there is no easy way to tell when an overflow will occur. As a guideline, if your source code occupies more than about 25K and contains calls to some really big macros then you may run into trouble but I think about 90% of the time you will be OK. Sorry about that!

I have tried to make the MACH system reasonably versatile by including the external command facility. If anyone has any comments, questions or ideas on the system, I would be glad to hear about them. Write to Steve Carnie, c/o Your Commodore.

```

10 4441:0F 442 7000 118
20 0-00000-0F10F INSTALLING 07040000*
30 FOR L=200 TO 900 STEP 10
40 T=0
50 FOR B=0 TO 2
60 READ 1:POKE 0,0-0-0-1-1-1-2
70 NEXT
80 READ T1:T1=T1+T000:PRINT"DATA ERROR IN LINE",L:GOTO
90 NEXT
10 PRINT"LOADING MONITOR."
110 LOAD"MONITOR",L:
120 POKE 0000,40
130 POKE 0000,0:POKE 0001,120
140 POKE 0001,120
150 POKE 0004,70:POKE 0005,144:POKE 0006,120
160 PRINT"SAVING MONITOR"
170 POKE 41,0:POKE 44,120:POKE 45,120:POKE 44,140
180 SAVE "MONITOR",L:
190 POKE 42,1:POKE 44,0-0-0
200 PRINT"FINISHED."
210 DATA 76,255,254,255,255,181,117,255,109,120,121,87,115,
220 255,255,41, 374
230 DATA 71,43,255,45,40,47, 33,49,255,10,255,4, 23,255,14,36,
240 1407
250 DATA 255,255,255,255,255, 76,255,255,44,255,180,255,
260 255,255, 3450
270 DATA 201,187,210,255,255,211,247,195,249,255,255,234,
280 229,255,255, 3467
290 DATA 255,255,255,255,255,255,182, 176,255,255,254,255,
300 255,255,255, 3747
310 DATA 255,255,198,214,255,254,255,255,255,255,255,71,
320 47,85,255, 3340
330 DATA 77,87,89,40,81,255,255,255,254,255,255,254,255,

```

```

340 255, 1120
350 DATA 255,255,255,255,255,255,74,155,255,255,155,154,255,
360 255,255, 1724
370 DATA 255,55,255,255,255,255,255,255,149,145,181,255,177,
380 187,182,141, 3795
390 DATA 177,255,255,163,144,255,182, 174,255,194,255,255,
400 255,255,144,144, 3405
410 DATA 186,255,175,188,255,255,255,255,74,255,70,86,255,
420 78,74,255, 2987
430 DATA 255,255,255,255,7, 5,11,255,15,25,25,1,17,255,41,
440 255, 1947
450 DATA 30,74,255,46,47,255,255,255,255,194,255,190,150,
460 255,115,175, 3747
470 DATA 255,255,255,255,255,255,255,193,255,255,255,247,
480 255,241,255,255, 3702
490 DATA 255,111,149,255,141,157,255,129,149,255,255,255,
500 154,255,150,142, 3740
510 DATA 255,255,255,255,255,255,111,148,255,194,255,
520 255,255,255, 3750
530 DATA 44,47,47,46,47, 83,44,47,47, 46,77,77,44,78,47,44,
540 1127
550 DATA 80,76,44,46,47, 46,44, 83,47,76,47,47,74,48,47,74,
560 1174
570 DATA 71,47,74,46,46, 49,46, 48,47, 87,71,70,88,71,78,47,
580 1003
590 DATA 80,71,45,80,71,80,80,76,45,80,76,80, 83,84,71,45,

```

```

600 1127
610 DATA 84,85,81,47,47,85,47,46, 83,47,71,84,45,88, 84,45,
620 1117
630 DATA 89,84, 81,88,84,88, 45,84, 88,45, 84,47, 45,75,79,84,
640 1111
650 DATA 44, 82,71,45,48, 47,47,78,48,45, 81,76,48,71,84,47,

```

1188

430 D478 71,80,87,88,89,87,88,87,87,87,79,82,77,78,  
1213

440 D478 87,79,77,80,74,82,82,76,88,85,76,88,76,88,87,  
1213

450 D478 76,81,82,79,82,82,82,79,74,82,79,82,82,84,87,87,  
1264

460 D478 84,82,82,84,88,82,84,87,84,87,88,88,88,88,84,88,  
1217, 1884

470 D478 84,81,88,184,202,126,272,288,75,8,87,88,87,88,84,  
348, 1978

480 D478 120,179,188,184,158,184,182,214,8,8,11,22,22,88,22,  
44, 1223

490 D478 77,88,97,118,121,121,145,184,182,178,187,178,287,  
228,21,282, 1222

500 D478 222,187,222,241,174,2,22,22,8,248,8,22,21,128,22,  
24, 1888

510 D478 128,22,18,128,12,228,222,248,228,201,12,228,241,  
188,2,2, 2884

520 D478 188,88,201,2,148,2,201,2,178,18,74,28,127,188,2,  
122, 1228

530 D478 76,76,76,127,288,8,188,18,288,18,288,24,188,2,122,  
91, 1292

540 D478 188,1,177,20,182,1,2,76,76,127,188,2,122,88,88,8,1,  
1212

550 D478 127,28,188,1,2,288,177,28,188,2,2,76,76,127,188,88,  
1284

560 D478 88,88,82,74,218,228,22,82,127,12,88,127,173,8,2,  
22, 1442

570 D478 12,128,188,91,224,1,248,18,173,1,2,22,12,128,188,  
91, 1488

580 D478 224,21,288,12,173,2,2,22,12,128,76,22,82,127,12,88,  
1228

590 D478 127,12,82,127,76,82,127,22,82,127,22,82,127,188,2,  
144, 1292

600 D478 8,128,24,18,182,2,178,188,8,128,22,228,222,222,288,  
192, 1884

610 D478 1,288,244,76,22,82,127,22,82,127,188,88,288,8,188,  
82, 1284

620 D478 22,228,222,187,1,76,221,2,178,88,201,1,288,18,187,  
22, 1828

630 D478 22,228,222,187,24,22,218,222,173,1,2,22,76,128,188,  
2, 1222

640 D478 76,187,24,22,218,222,173,1,2,22,76,128,188,88,201,  
2, 1888

650 D478 248,124,72,12,82,127,184,201,2,288,2,187,88,84,188,  
88, 1847

660 D478 22,228,222,187,2,76,221,8,178,88,72,187,24,22,218,  
222, 1988

670 D478 173,2,2,22,76,128,173,1,2,22,22,128,184,201,2,288,  
1222

680 D478 2,187,2,76,72,22,82,127,184,221,8,228,2,188,88,88,  
1287

690 D478 184,87,22,218,222,187,2,76,72,187,88,22,218,222,  
187,24, 2888

700 D478 22,228,222,244,221,18,248,88,72,173,2,2,22,76,128,  
184, 1822

710 D478 204,8,288,18,12,82,127,187,88,12,218,222,187,88,12,  
118, 2888

720 D478 222,187,2,76,188,81,22,218,222,22,82,127,187,88,12,  
118, 1928

730 D478 228,187,2,76,173,2,88,22,128,173,2,88,22,128,  
1297

740 D478 188,81,22,228,222,188,2,76,127,28,241,1,2,188,2,  
122, 1428

750 D478 91,22,76,127,22,177,127,22,82,127,12,82,127,188,28,  
24, 1222

760 D478 182,2,122,222,182,21,182,8,122,224,173,1,2,88,18,  
24, 1822

770 D478 281,222,127,222,182,224,182,8,122,224,74,178,128,  
184,2,24, 2228

780 D478 127,1,2,122,2,182,222,24,228,2,122,222,182,224,222,  
8, 2228

790 D478 122,228,188,224,22,22,128,182,22,22,128,74,28,  
104,24, 1828

800 D478 122,28,188,21,188,8,122,21,76,187,8,122,2,122,76,  
122, 1228

810 D478 91,22,82,127,182,21,22,82,128,188,28,72,188,88,  
82, 1217

820 D478 127,188,8,127,28,241,8,2,288,222,248,88,187,8,221,  
188, 1788

830 D478 128,248,8,222,224,22,244,228,74,76,128,124,2,224,88,  
174, 2877

840 D478 8,288,22,284,128,188,2,76,187,128,187,1,122,76,22,  
76, 1222

850 D478 127,12,127,127,187,8,76,187,128,187,1,122,76,22,76,  
127, 1291

860 D478 22,82,127,22,82,127,188,88,122,228,228,187,88,22,  
218,222, 1828

870 D478 188,84,22,228,222,22,82,127,22,82,127,188,24,22,  
218,222, 1888

880 D478 173,8,2,22,76,128,188,2,76,187,128,182,8,221,2,122,  
1884

890 D478 288,8,222,218,222,288,244,76,22,127,128,122,76,187,  
21,222, 2418

900 D478 188,128,184,2,244,8,222,224,22,288,244,222,124,2,  
182,76, 1213

910 D478 24,222,188,128,122,88,28,8,127,22,119,127,12,188,  
127,76, 1888

920 D478 187,128,228,228,228,222,222,222,222,222,222,222,  
222,222,222,222, 2887

930 D478 22,28,171,187,127,248,127,22,28,171,76,212,127,12,  
18,72, 1887

940 D478 82,82,72,88,22,88,22,88,88,88,88,88,88,88,88,88,  
1887

950 D478 82,22,18,88,84,88,76,82,72,79,78,12,18,88,87,82,  
1878

960 D478 82,72,79,78,12,21,88,22,82,88,84,12,18,88,87,  
888

970 D478 81,12,82,88,88,84,87,84,87,87,88,87,88,22,79,87,  
902

980 D478 84,79,88,88,82,82,88,87,24,22,12,18,8,8,222,128,  
1822



# A GIFT 12 TIMES A YEAR?

FOR JUST ...

# £13

NORMAL RATE ... £16.60

A subscription to **YOUR COMMODORE** magazine is a gift twelve times over! Whether the subscription is for a friend, relative or a treat for yourself, **YOUR COMMODORE** provides the reader with the latest information and developments on the Commodore range of computers.

Delivered each month straight to the comfort of your own home, could a Commodore enthusiast wish for anything better?

(This offer lasts until 24.2.85 and applies to readers in the United Kingdom only.)

#### SUBSCRIPTION RULES

Foreign overseas accelerated surface post: £21.80 or £33.00 D.O.  
Foreign overseas air mail: £57.00

Please return the completed form to:  
Your Commodore Special Offer  
c/o R4047 LTD  
Times House  
179 Fleet Street  
London EC4A 3DF  
Telex 8971 188

## Your COMMODORE

Please register the special £13.00 **YOUR COMMODORE** subscription below:

Name

Address

I would like to arrange a gift subscription to the person below:

(Enter your own name and address above as donor.)

Name

Address

Please commence the subscription to **YOUR COMMODORE** with the \_\_\_\_\_ issue.

I enclose payment of £

(Cheques must be payable to Argus Specialist Publications Ltd.)

Please charge my credit card account the amount of £

Card no.

Valid from  to

Signature

Name

Address

**This high speed tape operation for the CTR and Plus/4 will cut down that tedious waiting time. By Nick Hampshire.**

# BREAK THE SPEED LIMIT

A FAST LOADER IS A ROUTINE WHICH replaces the existing LOAD and allows a program or data to be loaded from tape at about 10 times the speed of a normal LOAD so a tape can be as fast as a disk drive.

A fast loader is achieved by simply changing the format of the pulse sequence stored on the tape to allow a far greater density of information storage per inch of tape.

In order to create a fast load program two routines are needed. Firstly, a fast LOAD routine. This is a fairly short machine code routine loaded at the beginning of a LOAD operation and autorun to LOAD the rest of the program and/or data stored in fast loader format. The second program required is a routine to SAVE a program in fast loader format: the fast SAVE routine.

The first major problem to be overcome in designing a fast loader is how to store each bit on the tape. Each bit is stored on tape as a pulse which goes through a high-low transition (see Figure 1). The length of the total pulse decides whether the bit is a 0 or 1. A short pulse is a 0 and a long pulse is a 1. The bit is flagged in the interrupt register on the falling edge of the pulse.

The loader is a machine code program which runs with the interrupts disabled, sets a timer between the two lengths, and when the timer runs out the interrupt register is checked to see if the pulse came in or not. If the falling edge of the pulse generates an interrupt before the timer runs out then the pulse was a zero, if not, it was a one. The bits are then rotated into a byte storage until eight bits have been read, thereby loading a full byte.

Before any bytes can be read and stored, the loader must set itself to be in sync with the bits on the tape. This is done by writing a string of 8 bits with a single 1 bit at every byte interval. The routine then tries to align itself by recognising the value of the byte. An example of a header byte for aligning would be the value 64, hex 40 or in binary 01000000. A series of these bytes is written as the header. Only when this byte has been read in and recognised can the actual program be read without risk of alignment errors.

The program is loaded in different ways depending on how much program protection is desired. The simplest way of formatting the file is to first SAVE the two byte load address followed by the two

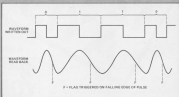


Figure 1

byte and address and then the actual file. The first byte following the end of the file is a checksum calculated by the SAVE routine and it's also calculated during loading. If the two values are the same, the LOAD was successful. The routine for this form of fast loader is given in Program 1.

## Fast tape routines — making them work

Putting the theory into practice to create the fast LOADER routine is not difficult. The actual timing for the SAVE routine was not calculated from any theoretical formula but was obtained merely by trial and error. The only guidelines were that the short pulse should be slightly shorter than half the long pulse, since the waveform of the pulse is skewed out by the cassette hardware. The timing value used by the loader is just shorter than the time required before the long pulse reaches its falling edge.

The high-speed tape routine will SAVE a Basic program to tape in fast format and automatically put the fast LOAD routine into the filename where it is stored and, when loaded, will automatically start on the warm start sector. The routines are initiated by SYS(10000). A Basic program can be fast saved by using the SAVE command as normal but with a device number of 7, thus:

```
SAVE"PROGRAM" 7
```

In addition the fast LOAD also makes use of the secondary address to autorun a program, thus:

```
SAVE"PROGRAM" 7,3
```

This will cause the program to autorun when loaded back. With both routines, when a program has both preceding one of these fast loader SAVE routines it is unnecessary to LOAD anything before the program; it will LOAD directly from the LOAD command.

## Program Listing 1

```

1000 833C      ;FAST TAPE SAVE FOR THE COMMODORE 16.
1010 833C      ;*****
1020 833C
1030 833C
1040 833C
1050 833C      ;THIS ROUTINE WILL SAVE A PROGRAM
1060 833C      ;TO TAPE SO THAT WHEN LOADED BACK
1070 833C      ;IT WILL LOAD EXACTLY AS FAST AS
1080 833C      ;THE 1541 DISK DRIVE.
1090 833C
1100 833C      ;AN OPTION FOR AUTO-RUN IS
1110 833C      ;INCLUDED.
1120 833C
1130 833C      *-#3000
1140 8300 R300      LBR #C03VEC      ;CHARGE SAVE VECTOR
1150 8302 830000      STA #R300      ; TO 160 TO 160
1160 8305 R300      LBR #C03VEC      ; SAVE ROUTINE
1170 8307 830100      STA #R301
1180 8309 80          RTS
1190 830A
1200 830B
1210 830C
1220 830D      ; HEDGE INTO THE SAVE COMMAND
1230 830E      ;CHECKS FOR DEVICE #7 AND IF SO,
1240 830F      ;SAVE IN FAST FORMAT.
1250 8310
1260 8311      40          SAVEC      FBR
1270 8312 85AC      LBR #RC
1280 8313 85AE      LBR #RC
1290 8314 85B7      CRR #R7
1300 8315 85B8      BSR TSAVE
1310 8316 85C0      PLA
1320 8317 85C1      JBR #P160
1330 8318
1340 8319      ; SAVE THE FILE. FIRST THE AUTO
1350 831A      ;LOAD ROUTINE IS SAVED IN NORMAL
1360 831B      ;TAPE FORMAT FOLLOWED BY THE FILE
1370 831C      ;IN FAST FORMAT.
1380 831D
1390 831E      TSAVE      LBR #R2
1400 831F      STA STORE
1410 8320 85D0      LBR #R3
1420 8321 85D1      STA STORE+1
1430 8322 85D2      LBR #R3
1440 8323 85D3      STA STORE+2
1450 8324 85D4      LBR #R4
1460 8325 85D5      STA STORE+3
1470 8326 85D6      LBR #R5
1480 8327 85D7      STA STORE+4
1490 8328 85D8      LBR #R6
1500 8329 85D9      STA STORE+5
1510 832A 85DA      LBR #R7
1520 832B 85DB      STA STORE+6
1530 832C 85DC      LBR #R8
1540 832D 85DD      STA STORE+7
1550 832E 85DE      LBR #R9
1560 832F 85DF      STA STORE+8
1570 8330 85E0      LBR #R10
1580 8331 85E1      STA STORE+9
1590 8332 85E2      LBR #R11
1600 8333 85E3      STA STORE+10
1610 8334 85E4      LBR #R12
1620 8335 85E5      STA STORE+11
1630 8336 85E6      LBR #R13
1640 8337 85E7      STA STORE+12
1650 8338 85E8      LBR #R14
1660 8339 85E9      STA STORE+13
1670 833A 85EA      LBR #R15
1680 833B 85EB      STA STORE+14
1690 833C 85EC      LBR #R16
1700 833D 85ED      STA STORE+15
1710 833E 85EE      LBR #R17
1720 833F 85EF      STA STORE+16
1730 8340 85F0      LBR #R18
1740 8341 85F1      STA STORE+17
1750 8342 85F2      LBR #R19
1760 8343 85F3      STA STORE+18
1770 8344 85F4      LBR #R20
1780 8345 85F5      STA STORE+19
1790 8346 85F6      LBR #R21
1800 8347 85F7      STA STORE+20
1810 8348 85F8      LBR #R22
1820 8349 85F9      STA STORE+21
1830 834A 85FA      LBR #R23
1840 834B 85FB      STA STORE+22
1850 834C 85FC      LBR #R24
1860 834D 85FD      STA STORE+23
1870 834E 85FE      LBR #R25
1880 834F 85FF      STA STORE+24
1890 8350 8600      LBR #R26
1900 8351 8601      STA STORE+25
1910 8352 8602      LBR #R27
1920 8353 8603      STA STORE+26
1930 8354 8604      LBR #R28
1940 8355 8605      STA STORE+27
1950 8356 8606      LBR #R29
1960 8357 8607      STA STORE+28
1970 8358 8608      LBR #R30
1980 8359 8609      STA STORE+29
1990 835A 860A      LBR #R31
2000 835B 860B      STA STORE+30
2010 835C 860C      LBR #R32
2020 835D 860D      STA STORE+31
2030 835E 860E      LBR #R33
2040 835F 860F      STA STORE+32
2050 8360 8610      LBR #R34
2060 8361 8611      STA STORE+33
2070 8362 8612      LBR #R35
2080 8363 8613      STA STORE+34
2090 8364 8614      LBR #R36
2100 8365 8615      STA STORE+35
2110 8366 8616      LBR #R37
2120 8367 8617      STA STORE+36
2130 8368 8618      LBR #R38
2140 8369 8619      STA STORE+37
2150 836A 861A      LBR #R39
2160 836B 861B      STA STORE+38
2170 836C 861C      LBR #R40
2180 836D 861D      STA STORE+39
2190 836E 861E      LBR #R41
2200 836F 861F      STA STORE+40
2210 8370 8620      LBR #R42
2220 8371 8621      STA STORE+41
2230 8372 8622      LBR #R43
2240 8373 8623      STA STORE+42
2250 8374 8624      LBR #R44
2260 8375 8625      STA STORE+43
2270 8376 8626      LBR #R45
2280 8377 8627      STA STORE+44
2290 8378 8628      LBR #R46
2300 8379 8629      STA STORE+45
2310 837A 862A      LBR #R47
2320 837B 862B      STA STORE+46
2330 837C 862C      LBR #R48
2340 837D 862D      STA STORE+47
2350 837E 862E      LBR #R49
2360 837F 862F      STA STORE+48
2370 8380 8630      LBR #R50
2380 8381 8631      STA STORE+49
2390 8382 8632      LBR #R51
2400 8383 8633      STA STORE+50
2410 8384 8634      LBR #R52
2420 8385 8635      STA STORE+51
2430 8386 8636      LBR #R53
2440 8387 8637      STA STORE+52
2450 8388 8638      LBR #R54
2460 8389 8639      STA STORE+53
2470 838A 863A      LBR #R55
2480 838B 863B      STA STORE+54
2490 838C 863C      LBR #R56
2500 838D 863D      STA STORE+55
2510 838E 863E      LBR #R57
2520 838F 863F      STA STORE+56
2530 8390 8640      LBR #R58
2540 8391 8641      STA STORE+57
2550 8392 8642      LBR #R59
2560 8393 8643      STA STORE+58
2570 8394 8644      LBR #R60
2580 8395 8645      STA STORE+59
2590 8396 8646      LBR #R61
2600 8397 8647      STA STORE+60
2610 8398 8648      LBR #R62
2620 8399 8649      STA STORE+61
2630 839A 864A      LBR #R63
2640 839B 864B      STA STORE+62
2650 839C 864C      LBR #R64
2660 839D 864D      STA STORE+63
2670 839E 864E      LBR #R65
2680 839F 864F      STA STORE+64
2690 83A0 8650      LBR #R66
2700 83A1 8651      STA STORE+65
2710 83A2 8652      LBR #R67
2720 83A3 8653      STA STORE+66
2730 83A4 8654      LBR #R68
2740 83A5 8655      STA STORE+67
2750 83A6 8656      LBR #R69
2760 83A7 8657      STA STORE+68
2770 83A8 8658      LBR #R70
2780 83A9 8659      STA STORE+69
2790 83AA 865A      LBR #R71
2800 83AB 865B      STA STORE+70
2810 83AC 865C      LBR #R72
2820 83AD 865D      STA STORE+71
2830 83AE 865E      LBR #R73
2840 83AF 865F      STA STORE+72
2850 83B0 8660      LBR #R74
2860 83B1 8661      STA STORE+73
2870 83B2 8662      LBR #R75
2880 83B3 8663      STA STORE+74
2890 83B4 8664      LBR #R76
2900 83B5 8665      STA STORE+75
2910 83B6 8666      LBR #R77
2920 83B7 8667      STA STORE+76
2930 83B8 8668      LBR #R78
2940 83B9 8669      STA STORE+77
2950 83BA 866A      LBR #R79
2960 83BB 866B      STA STORE+78
2970 83BC 866C      LBR #R80
2980 83BD 866D      STA STORE+79
2990 83BE 866E      LBR #R81
3000 83BF 866F      STA STORE+80
3010 83C0 8670      LBR #R82
3020 83C1 8671      STA STORE+81
3030 83C2 8672      LBR #R83
3040 83C3 8673      STA STORE+82
3050 83C4 8674      LBR #R84
3060 83C5 8675      STA STORE+83
3070 83C6 8676      LBR #R85
3080 83C7 8677      STA STORE+84
3090 83C8 8678      LBR #R86
3100 83C9 8679      STA STORE+85
3110 83CA 867A      LBR #R87
3120 83CB 867B      STA STORE+86
3130 83CC 867C      LBR #R88
3140 83CD 867D      STA STORE+87
3150 83CE 867E      LBR #R89
3160 83CF 867F      STA STORE+88
3170 83D0 8680      LBR #R90
3180 83D1 8681      STA STORE+89
3190 83D2 8682      LBR #R91
3200 83D3 8683      STA STORE+90
3210 83D4 8684      LBR #R92
3220 83D5 8685      STA STORE+91
3230 83D6 8686      LBR #R93
3240 83D7 8687      STA STORE+92
3250 83D8 8688      LBR #R94
3260 83D9 8689      STA STORE+93
3270 83DA 868A      LBR #R95
3280 83DB 868B      STA STORE+94
3290 83DC 868C      LBR #R96
3300 83DD 868D      STA STORE+95
3310 83DE 868E      LBR #R97
3320 83DF 868F      STA STORE+96
3330 83E0 8690      LBR #R98
3340 83E1 8691      STA STORE+97
3350 83E2 8692      LBR #R99
3360 83E3 8693      STA STORE+98
3370 83E4 8694      LBR #R100
3380 83E5 8695      STA STORE+99
3390 83E6 8696      LBR #R101
3400 83E7 8697      STA STORE+100
3410 83E8 8698      LBR #R102
3420 83E9 8699      STA STORE+101
3430 83EA 869A      LBR #R103
3440 83EB 869B      STA STORE+102
3450 83EC 869C      LBR #R104
3460 83ED 869D      STA STORE+103
3470 83EE 869E      LBR #R105
3480 83EF 869F      STA STORE+104
3490 83F0 86A0      LBR #R106
3500 83F1 86A1      STA STORE+105
3510 83F2 86A2      LBR #R107
3520 83F3 86A3      STA STORE+106
3530 83F4 86A4      LBR #R108
3540 83F5 86A5      STA STORE+107
3550 83F6 86A6      LBR #R109
3560 83F7 86A7      STA STORE+108
3570 83F8 86A8      LBR #R110
3580 83F9 86A9      STA STORE+109
3590 83FA 86AA      LBR #R111
3600 83FB 86AB      STA STORE+110
3610 83FC 86AC      LBR #R112
3620 83FD 86AD      STA STORE+111
3630 83FE 86AE      LBR #R113
3640 83FF 86AF      STA STORE+112
3650 8400 86B0      LBR #R114
3660 8401 86B1      STA STORE+113
3670 8402 86B2      LBR #R115
3680 8403 86B3      STA STORE+114
3690 8404 86B4      LBR #R116
3700 8405 86B5      STA STORE+115
3710 8406 86B6      LBR #R117
3720 8407 86B7      STA STORE+116
3730 8408 86B8      LBR #R118
3740 8409 86B9      STA STORE+117
3750 840A 86BA      LBR #R119
3760 840B 86BB      STA STORE+118
3770 840C 86BC      LBR #R120
3780 840D 86BD      STA STORE+119
3790 840E 86BE      LBR #R121
3800 840F 86BF      STA STORE+120
3810 8410 86C0      LBR #R122
3820 8411 86C1      STA STORE+121
3830 8412 86C2      LBR #R123
3840 8413 86C3      STA STORE+122
3850 8414 86C4      LBR #R124
3860 8415 86C5      STA STORE+123
3870 8416 86C6      LBR #R125
3880 8417 86C7      STA STORE+124
3890 8418 86C8      LBR #R126
3900 8419 86C9      STA STORE+125
3910 841A 86CA      LBR #R127
3920 841B 86CB      STA STORE+126
3930 841C 86CC      LBR #R128
3940 841D 86CD      STA STORE+127
3950 841E 86CE      LBR #R129
3960 841F 86CF      STA STORE+128
3970 8420 86D0      LBR #R130
3980 8421 86D1      STA STORE+129
3990 8422 86D2      LBR #R131
4000 8423 86D3      STA STORE+130
4010 8424 86D4      LBR #R132
4020 8425 86D5      STA STORE+131
4030 8426 86D6      LBR #R133
4040 8427 86D7      STA STORE+132
4050 8428 86D8      LBR #R134
4060 8429 86D9      STA STORE+133
4070 842A 86DA      LBR #R135
4080 842B 86DB      STA STORE+134
4090 842C 86DC      LBR #R136
4100 842D 86DD      STA STORE+135
4110 842E 86DE      LBR #R137
4120 842F 86DF      STA STORE+136
4130 8430 86E0      LBR #R138
4140 8431 86E1      STA STORE+137
4150 8432 86E2      LBR #R139
4160 8433 86E3      STA STORE+138
4170 8434 86E4      LBR #R140
4180 8435 86E5      STA STORE+139
4190 8436 86E6      LBR #R141
4200 8437 86E7      STA STORE+140
4210 8438 86E8      LBR #R142
4220 8439 86E9      STA STORE+141
4230 843A 86EA      LBR #R143
4240 843B 86EB      STA STORE+142
4250 843C 86EC      LBR #R144
4260 843D 86ED      STA STORE+143
4270 843E 86EE      LBR #R145
4280 843F 86EF      STA STORE+144
4290 8440 86F0      LBR #R146
4300 8441 86F1      STA STORE+145
4310 8442 86F2      LBR #R147
4320 8443 86F3      STA STORE+146
4330 8444 86F4      LBR #R148
4340 8445 86F5      STA STORE+147
4350 8446 86F6      LBR #R149
4360 8447 86F7      STA STORE+148
4370 8448 86F8      LBR #R150
4380 8449 86F9      STA STORE+149
4390 844A 86FA      LBR #R151
4400 844B 86FB      STA STORE+150
4410 844C 86FC      LBR #R152
4420 844D 86FD      STA STORE+151
4430 844E 86FE      LBR #R153
4440 844F 86FF      STA STORE+152
4450 8450 8700      LBR #R154
4460 8451 8701      STA STORE+153
4470 8452 8702      LBR #R155
4480 8453 8703      STA STORE+154
4490 8454 8704      LBR #R156
4500 8455 8705      STA STORE+155
4510 8456 8706      LBR #R157
4520 8457 8707      STA STORE+156
4530 8458 8708      LBR #R158
4540 8459 8709      STA STORE+157
4550 845A 870A      LBR #R159
4560 845B 870B      STA STORE+158
4570 845C 870C      LBR #R160
4580 845D 870D      STA STORE+159
4590 845E 870E      LBR #R161
4600 845F 870F      STA STORE+160
4610 8460 8710      LBR #R162
4620 8461 8711      STA STORE+161
4630 8462 8712      LBR #R163
4640 8463 8713      STA STORE+162
4650 8464 8714      LBR #R164
4660 8465 8715      STA STORE+163
4670 8466 8716      LBR #R165
4680 8467 8717      STA STORE+164
4690 8468 8718      LBR #R166
4700 8469 8719      STA STORE+165
4710 846A 871A      LBR #R167
4720 846B 871B      STA STORE+166
4730 846C 871C      LBR #R168
4740 846D 871D      STA STORE+167
4750 846E 871E      LBR #R169
4760 846F 871F      STA STORE+168
4770 8470 8720      LBR #R170
4780 8471 8721      STA STORE+169
4790 8472 8722      LBR #R171
4800 8473 8723      STA STORE+170
4810 8474 8724      LBR #R172
4820 8475 8725      STA STORE+171
4830 8476 8726      LBR #R173
4840 8477 8727      STA STORE+172
4850 8478 8728      LBR #R174
4860 8479 8729      STA STORE+173
4870 847A 872A      LBR #R175
4880 847B 872B      STA STORE+174
4890 847C 872C      LBR #R176
4900 847D 872D      STA STORE+175
4910 847E 872E      LBR #R177
4920 847F 872F      STA STORE+176
4930 8480 8730      LBR #R178
4940 8481 8731      STA STORE+177
4950 8482 8732      LBR #R179
4960 8483 8733      STA STORE+178
4970 8484 8734      LBR #R180
4980 8485 8735      STA STORE+179
4990 8486 8736      LBR #R181
5000 8487 8737      STA STORE+180
5010 8488 8738      LBR #R182
5020 8489 8739      STA STORE+181
5030 848A 873A      LBR #R183
5040 848B 873B      STA STORE+182
5050 848C 873C      LBR #R184
5060 848D 873D      STA STORE+183
5070 848E 873E      LBR #R185
5080 848F 873F      STA STORE+184
5090 8490 8740      LBR #R186
5100 8491 8741      STA STORE+185
5110 8492 8742      LBR #R187
5120 8493 8743      STA STORE+186
5130 8494 8744      LBR #R188
5140 8495 8745      STA STORE+187
5150 8496 8746      LBR #R189
5160 8497 8747      STA STORE+188
5170 8498 8748      LBR #R190
5180 8499 8749      STA STORE+189
5190 849A 874A      LBR #R191
5200 849B 874B      STA STORE+190
5210 849C 874C      LBR #R192
5220 849D 874D      STA STORE+191
5230 849E 874E      LBR #R193
5240 849F 874F      STA STORE+192
5250 84A0 8750      LBR #R194
5260 84A1 8751      STA STORE+193
5270 84A2 8752      LBR #R195
5280 84A3 8753      STA STORE+194
5290 84A4 8754      LBR #R196
5300 84A5 8755      STA STORE+195
5310 84A6 8756      LBR #R197
5320 84A7 8757      STA STORE+196
5330 84A8 8758      LBR #R198
5340 84A9 8759      STA STORE+197
5350 84AA 875A      LBR #R199
5360 84AB 875B      STA STORE+198
5370 84AC 875C      LBR #R200
5380 84AD 875D      STA STORE+199
5390 84AE 875E      LBR #R201
5400 84AF 875F      STA STORE+200
5410 84B0 8760      LBR #R202
5420 84B1 8761      STA STORE+201
5430 84B2 8762      LBR #R203
5440 84B3 8763      STA STORE+202
5450 84B4 8764      LBR #R204
5460 84B5 8765      STA STORE+203
5470 84B6 8766      LBR #R205
5480 84B7 8767      STA STORE+204
5490 84B8 8768      LBR #R206
5500 84B9 8769      STA STORE+205
5510 84BA 876A      LBR #R207
5520 84BB 876B      STA STORE+206
5530 84BC 876C      LBR #R208
5540 84BD 876D      STA STORE+207
5550 84BE 876E      LBR #R209
5560 84BF 876F      STA STORE+208
5570 84C0 8770      LBR #R210
5580 84C1 8771      STA STORE+209
5590 84C2 8772      LBR #R211
5600 84C3 8773      STA STORE+210
5610 84C4 8774      LBR #R212
5620 84C5 8775      STA STORE+211
5630 84C6 8776      LBR #R213
5640 84C7 8777      STA STORE+212
5650 84C8 8778      LBR #R214
5660 84C9 8779      STA STORE+213
5670 84CA 877A      LBR #R215
5680 84CB 877B      STA STORE+214
5690 84CC 877C      LBR #R216
5700 84CD 877D      STA STORE+215
5710 84CE 877E      LBR #R217
5720 84CF 877F      STA STORE+216
5730 84D0 8780      LBR #R218
5740 84D1 8781      STA STORE+217
5750 84D2 8782      LBR #R219
5760 84D3 8783      STA STORE+218
5770 84D4 8784      LBR #R220
5780 84D5 8785      STA STORE+219
5790 84D6 8786      LBR #R221
5800 84D7 8787      STA STORE+220
5810 84D8 8788      LBR #R222
5820 84D9 8789      STA STORE+221
5830 84DA 878A      LBR #R223
5840 84DB 878B      STA STORE+222
5850 84DC 878C      LBR #R224
5860 84DD 878D      STA STORE+223
5870 84DE 878E      LBR #R225
5880 84DF 878F      STA STORE+224
5890 84E0 8790      LBR #R226
5900 84E1 8791      STA STORE+225
5910 84E2 8792      LBR #R227
5920 84E3 8793      STA STORE+226
5930 84E4 8794      LBR #R228
5940 84E5 8795      STA STORE+227
5950 84E6 8796      LBR #R229
5960 84E7 8797      STA STORE+228
5970 84E8 8798      LBR #R230
5980 84E9 8799      STA STORE+229
5990 84EA 879A      LBR #R231
6000 84EB 879B      STA STORE+230
6010 84EC 879C      LBR #R232
6020 84ED 879D      STA STORE+231
6030 84EE 879E      LBR #R233
6040 84EF 879F      STA STORE+232
6050 84F0 87A0      LBR #R234
6060 84F1 87A1      STA STORE+233
6070 84F2 87A2      LBR #R235
6080 84F3 87A3      STA STORE+234
6090 84F4 87A4      LBR #R236
6100 84F5 87A5      STA STORE+235
6110 84F6 87A6      LBR #R237
6120 84F7 87A7      STA STORE+236
6130 84F8 87A8      LBR #R238
6140 84F9 87A9      STA STORE+237
6150 84FA 87AA      LBR #R239
6160 84FB 87AB      STA STORE+238
6170 84FC 87AC      LBR #R240
6180 84FD 87AD      STA STORE+239
6190 84FE 87AE      LBR #R241
6200 84FF 87AF      STA STORE+240
6210 8500 87B0      LBR #R242
6220 8501 87B1      STA STORE+241
6230 8502 87B2      LBR #R243
6240 8503 87B3      STA STORE+242
6250 8504 87B4      LBR #R244
6260 8505 87B5      STA STORE+243
6270 8506 87B6      LBR #R245
6280 8507 87B7      STA STORE+244
6290 8508 87B8      LBR #R246
6300 8509 87B9      STA STORE+245
6310 850A 87BA      LBR #R247
6320 850B 87BB      STA STORE+246
6330 850C 87BC      LBR #R248
6340 850D 87BD      STA STORE+247
6350 850E 87BE      LBR #R249
6360 850F 87BF      STA STORE+248
6370 8510 87C0      LBR #R250
6380 8511 87C1      STA STORE+249
6390 8512 87C2      LBR #R251
6400 8513 87C3      STA STORE+250
6410 8514 87C4      LBR #R252
6420 8515 87C5      STA STORE+251
6430 8516 87C6      LBR #R253
6440 8517 87C7      STA STORE+252
6450 8518 87C8      LBR #R254
6460 8519 87C9      STA STORE+253
6470 851A 87CA      LBR #R255
6480 851B 87CB      STA STORE+254
6490 851C 87CC      LBR #R256
6500 851D 87CD      STA STORE+255
6510 851E 87CE      LBR #R257
6520 851F 87CF      STA STORE+256
6530 8520 87D0      LBR #R258
6540 8521 87D1      STA STORE+257
6550 8522 87D2      LBR #R259
6560 8523 87D3      STA STORE+258
6570 8524 87D4      LBR #R260
6580 8525 87D5      STA STORE+259
6590 8526 87D6      LBR #R261
6600 8527 87D7      STA STORE+260
6610 8528 87D8      LBR #R262
6620 8529 87D9      STA STORE+261
6630 852A 87DA      LBR #R263
6640 852B 87DB      STA STORE+262
6650 852C 87DC      LBR #R264
6660 852D 87DD      STA STORE+263
6670 852E 87DE      LBR #R265
6680 852F 87DF      STA STORE+264
6690 8530 87E0      LBR #R266
6700 8531 87E1      STA STORE+265
6710 8532 
```

# Program Listing 1 (cont.)

```

1600 3361 802543      STA #8025
1610 3364 8001      LDA #801
1700 3366 80      TAX
1710 3367 80      TAX
1720 3368 2000FF      JSR #FF00          ISET FILE DETAILS
1730 3369 800C      LDA #80C
1740 3370 8000      LDC #0FLAME
1750 3371 8000      LDY #0FLAME
1760 3371 2000FF      JSR #FF00          ISET FILENAME DETAILS
1770 3374 8000      LDA #800
1780 3375 8000      STA #80
1790 3376 8000      LDA #80
1800 3377 8000      STA #80
1810 3378 8000      STA #80
1820 3379 8000      LDA #80
1830 3380 8000      STA #80
1840 3381 8000      LDA #80
1850 3382 8000      LDY #80
1860 3383 8000      LDY #80
1870 3384 8000      JSR #FF00          ISAVE IT
1880 3385 2000FF      LDA #80
1890 3386 8000      STA #80
1900 3387 8000      STA #80
1910 3388 8000      STA #80
1920 3389 8000      STA #80
1930 3390 8000      LDA #80
1940 3391 8000      TAX
1950 3392 8000      TAX
1960 3393 8000      STA #80
1970 3394 8000      LDA #80
1980 3395 8000      STA #80
1990 3396 8000      STA #80
2000 3397 8000      LDA #80
2010 3398 8000      STA #80
2020 3399 8000      STA #80
2030 3400 8000      LDA #80
2040 3401 8000      LDY #80
2050 3402 8000      LDA #80
2060 3403 8000      LDA #80
2070 3404 8000      LDA #80
2080 3405 8000      LDA #80
2090 3406 8000      LDA #80
2100 3407 8000      LDA #80
2110 3408 8000      LDA #80
2120 3409 8000      LDA #80
2130 3410 8000      LDA #80
2140 3411 8000      LDA #80
2150 3412 8000      LDA #80
2160 3413 8000      LDA #80
2170 3414 8000      LDA #80
2180 3415 8000      LDA #80
2190 3416 8000      LDA #80
2200 3417 8000      LDA #80
2210 3418 8000      LDA #80
2220 3419 8000      LDA #80
2230 3420 8000      LDA #80
2240 3421 8000      LDA #80
2250 3422 8000      LDA #80
2260 3423 8000      LDA #80
2270 3424 8000      LDA #80
2280 3425 8000      LDA #80
2290 3426 8000      LDA #80
2300 3427 8000      LDA #80
2310 3428 8000      LDA #80
2320 3429 8000      LDA #80
2330 3430 8000      LDA #80
2340 3431 8000      LDA #80
2350 3432 8000      LDA #80

```

## Program Listing 1 (cont.)

```

2360 20F3 C490 TSMV03      OPV #00
2370 20F3 A5B0             LDA #B0
2380 20F3 C390             SEC #00
2390 20F7 A6B0             BCC TSMVLOOP
2400 20F9 A5B4             LDA #B4
2410 20F9 204400          JSR WRTBYT
2420 20F1 200400          JSR WRTBYT
2430 2001 F010             LDA #A0
2440 2003 0000FF          STA #FF00
2450 2006 0000FF          STA #FF00
2460 2009 A000             LDA #0000
2470 200B 0000             STA #00
2480 200D 50             CLD
2490 200E 2004FF          JSR #FF04
2500 2011 00             RTS
2510 2410
2520 2013 70
2530 2013 0000FF          WRTBYT STA #FF0F
2540 2016 F000             LDA #0000
2550 2018 0000             STA #00
2560 201A F000             LDA #0000
2570 201C 0000FF          STA #FF00
2580 201F 00             DEC #00
2590 2020 00FD             BNE HENDR1
2600 2022 00             BEY
2610 2023 20FA             BNE HENDR1
2620 2025 F000             LDA #0000
2630 2027 0000FF          STA #FF00
2640 2029 F000             LDA #0000
2650 202C 0000FF          STA #FF00
2660 202F 0010             LDA #0010
2670 2031 0000FF          STA #FF00
2680 2034 F000             LDY #0000
2690 2036 F010             LDA #0010
2700 2038 204400          JSR WRTBYT
2710 203B 00             BEY
2720 203C 20FA             BNE HENDR2
2730 203E 0000             LDA #0000
2740 2040
2750 2040 0000          WRTBYT STA #00
2760 2042 4004             EOR #04
2770 2044 0004             STA #04
2780 2046 F000             LDA #0000
2790 2048 0000             STA #00
2800 204A 2000             ROL #00
2810 204C 200400          JSR WRTBYT
2820 204F 0000             SEC #00
2830 2051 00F7             BNE WRTBYT
2840 2053 00             RTS
2850 2054
2860 2054 A000          WRTBYT LDA #00
2870 2056 0000             BCC WRTT3
2880 2058 0000             LDA #00FF
2890 205A 200000          WRTT3 JSR WRTT3
2900 205D
2910 205D F010          WRTT3 LDA #0010
2920 205F 2000FF          BIT #FF00
2930 2062 F0F0             BNE WRTT3
2940 2064 40             PUL
2950 2066 F000             LDA #0000
2960 2068 0000FF          STA #FF00
2970 206A 0000FF          STA #FF00
2980 206C 00             PLS
2990 206E 0000FF          STA #FF00
3000 2071 0000             LDA #00
3010 2073 4000             EOR #00
3020 2075 0000             STA #00
3030 2077 00             RTS

```

# Program Listing 1 (cont.)

3040 3070	THE LOADER STARTS HERE		
3050 3070			
3060 3070			
3070 3070 0040	LOADER	LDR #040	1SET RESET VECTOR
3080 3070 00FFC		STR #FFFC	
3090 3070 0000		LDR #000	
3100 3070 00FFF		STR #FFFF	
3110 3080 00FFF		STR #FFF	FROM OUT
3120 3080 001FF		LDR #FF10	INCREASE BORDER COLOUR
3130 3080 10		CLC	ITELL LOADER TO 'APP'
3140 3080 004F0		JSR #004F	LOAD R FILE
3150 3080 000FF		STR #FF0	FROM IN
3160 3080 000		LDR #0	
3170 3091 001FF		STR #FF10	RESET BORDER
3180 3094 00		CLT	START I/O
3190 3095 000FF		JSR #FF00	RESET OUTPUT VECTOR
3200 3090 0004F		JSR #FF04	RESET I/O
3210 3090 000C		LDR #0C	ICMPARE CHECKSUM
3220 3090 0000		CPY #00	
3230 3094 0010		BNE L00000	DIFFERENT-LOAD ERROR
3240 3091 004F0		LDR #004F	RUN FLAG
3250 3094 0000		MOV #011	LOAD, RETURN TO READY
3260 3094 00FF		LDR #FF	TURN OFF CURSOR
3270 3090 000FF		STR #FF0C	JUST IN CASE
3280 3090 000FF		STR #FF00	
3290 3090 00000		JSR #0000	PERFORM 'RUN'
3300 3091 40000		JMP #0000	EXECUTE NEXT STATEMENT
3310 3094			
3320 3094 00000 011		JSR #0000	PERFORM 'CLR'
3330 3097 40000		JMP #0700	'READY.'
3340 3094			
3350 3094 0010 L00000		LDR #0010	# FOR 'LOAD'
3360 3090 40000		JMP #0000	OUTPUT ERROR
3370 3097			
3380 3097 00		BVT 0	GO TO RUN PROGRAM
3390 3090			
3400 3090		#=00007	
3410 3090			
3420 3090 000000	FLANGE	MOV #	
3430 3091	17 SPACES		
3440 3091	#=00040		
3450 3091 001FF		LDR #FF10	SAVE BORDER COLOUR
3460 3094 000		STR #0	
3470 3090 00		MOV	ITELL LOADER TO 'APP'
3480 3097 70		SET	DESABLE I/O
3490 3090	#=0004F		
3500 3090 00		MOV	SAVE STATUS
3510 3090 00000		JSR #0000	READ THE HEADER
3520 3090 00000		JSR #0000	READ LOAD LOW
3530 3097 00		MOV	LOAD .V
3540 3090 000		LDR #000	LOAD LOW BYTE
3550 3097 000		STR #0	
3560 3094 00000		JSR #0000	READ LOW HIGH
3570 3097 000		STR #0	
3580 3097 00000		JSR #0000	READ END LOW
3590 3097 000		STR #0	
3600 3097 00000		JSR #0000	READ END HIGH
3610 3097 000		STR #0	
3620 3097 00000 TLOAD0		JSR #0000	READ BYTE
3630 3097 000		STR #000.V	STORE IT
3640 3097 400C		MOV #0C	CALCULATE CHECKSUM
3650 3097 000C		STR #0C	
3660 3097 00		MOV	INCREMENT POINTER
3670 3097 0000		BNE TLOAD0	NOT NEW PAGE
3680 3097 000		MOV #0	INCREMENT HIGH BYTE
3690 3091 001FF		STR #FF10	INCREASE BORDER COLOUR
3700 3094 000	TLOAD0	CPY #0	END OF FILE?
3710 3090 000		LDR #0	



```

1000 ROM *****
1010 ROM * FIRST SAVE ROUTINE FOR THE
1020 ROM * MICRODORE 16. THIS ROUTINE
1030 ROM * SAVED UP ONLY 700 BYTES OF
1040 ROM * BASIC MEMORY.
1050 ROM *
1060 ROM * COPYRIGHT 1983
1070 ROM * ZIFAR SOFTWARE LTD.
1080 ROM *
1090 ROM *****
1100 POKE 51.0:POKE 53.0:POKE 55.0:POKE 57.0:POKE 59.0:POKE 61.0:CLR
1110 1-DEC("0000")-0
1120 READ # 17 #=1 THEN 1150
1130 POKE # 1-#+1#
1140 1-#-1-#010 1120
1150 1-#-1-#010 1120
1160 IF 1-#010 THEN PRINT"CHECKOUT ERROR :""SHOULD BE 0400" AND
1170 IF 1-#010 THEN PRINT"NUMBER OF VALUES ERROR :""SHOULD BE 0400"
1180
1190 PRINT"DATA ENTERED CORRECTLY."
1200 PRINT"TO TEST SAVE A BASIC PROGRAM USE:"
1210 PRINT"SAVE CHR$(34)"FILENAME "CHR$(34)".7"
1220 PRINT"OR MOVE CHR$(34)"FILENAME "CHR$(34)".7.1 TO AUTO-DEL."
1230
1240 SYS DEK("0000")-0
1250
1260 DATA 100,101,102,103,104,105,106,107,
1270 DATA 108,109,110,111,112,113,114,115,
1280 DATA 116,117,118,119,120,121,122,123,
1290 DATA 124,125,126,127,128,129,130,131,
1300 DATA 132,133,134,135,136,137,138,139,
1310 DATA 140,141,142,143,144,145,146,147,
1320 DATA 148,149,150,151,152,153,154,155,
1330 DATA 156,157,158,159,160,161,162,163,
1340 DATA 164,165,166,167,168,169,170,171,
1350 DATA 172,173,174,175,176,177,178,179,
1360 DATA 180,181,182,183,184,185,186,187,
1370 DATA 188,189,190,191,192,193,194,195,
1380 DATA 196,197,198,199,200,201,202,203,
1390 DATA 204,205,206,207,208,209,210,211,
1400 DATA 212,213,214,215,216,217,218,219,
1410 DATA 220,221,222,223,224,225,226,227,
1420 DATA 228,229,230,231,232,233,234,235,
1430 DATA 236,237,238,239,240,241,242,243,
1440 DATA 244,245,246,247,248,249,250,251,
1450 DATA 252,253,254,255,256,257,258,259,
1460 DATA 260,261,262,263,264,265,266,267,
1470 DATA 268,269,270,271,272,273,274,275,
1480 DATA 276,277,278,279,280,281,282,283,
1490 DATA 284,285,286,287,288,289,290,291,
1500 DATA 292,293,294,295,296,297,298,299,
1510 DATA 300,301,302,303,304,305,306,307,
1520 DATA 308,309,310,311,312,313,314,315,
1530 DATA 316,317,318,319,320,321,322,323,
1540 DATA 324,325,326,327,328,329,330,331,
1550 DATA 332,333,334,335,336,337,338,339,
1560 DATA 340,341,342,343,344,345,346,347,
1570 DATA 348,349,350,351,352,353,354,355,
1580 DATA 356,357,358,359,360,361,362,363,
1590 DATA 364,365,366,367,368,369,370,371,
1600 DATA 372,373,374,375,376,377,378,379,
1610 DATA 380,381,382,383,384,385,386,387,
1620 DATA 388,389,390,391,392,393,394,395,
1630 DATA 396,397,398,399,400,401,402,403,
1640 DATA 404,405,406,407,408,409,410,411,
1650 DATA 412,413,414,415,416,417,418,419,
1660 DATA 420,421,422,423,424,425,426,427,
1670 DATA 428,429,430,431,432,433,434,435,
1680 DATA 436,437,438,439,440,441,442,443,
1690 DATA 444,445,446,447,448,449,450,451,
1700 DATA 452,453,454,455,456,457,458,459,
1710 DATA 460,461,462,463,464,465,466,467,
1720 DATA 468,469,470,471,472,473,474,475,
1730 DATA 476,477,478,479,480,481,482,483,
1740 DATA 484,485,486,487,488,489,490,491,
1750 DATA 492,493,494,495,496,497,498,499,
1760 DATA 500,501,502,503,504,505,506,507,
1770 DATA 508,509,510,511,512,513,514,515,
1780 DATA 516,517,518,519,520,521,522,523,
1790 DATA 524,525,526,527,528,529,530,531,
1800 DATA 532,533,534,535,536,537,538,539,
1810 DATA 540,541,542,543,544,545,546,547,
1820 DATA 548,549,550,551,552,553,554,555,
1830 DATA 556,557,558,559,560,561,562,563,
1840 DATA 564,565,566,567,568,569,570,571,
1850 DATA 572,573,574,575,576,577,578,579,
1860 DATA 580,581,582,583,584,585,586,587,
1870 DATA 588,589,590,591,592,593,594,595,
1880 DATA 596,597,598,599,600,601,602,603,
1890 DATA 604,605,606,607,608,609,610,611,
1900 DATA 612,613,614,615,616,617,618,619,
1910 DATA 620,621,622,623,624,625,626,627,
1920 DATA 628,629,630,631,632,633,634,635,
1930 DATA 636,637,638,639,640,641,642,643,
1940 DATA 644,645,646,647,648,649,650,651,
1950 DATA 652,653,654,655,656,657,658,659,
1960 DATA 660,661,662,663,664,665,666,667,
1970 DATA 668,669,670,671,672,673,674,675,
1980 DATA 676,677,678,679,680,681,682,683,
1990 DATA 684,685,686,687,688,689,690,691,
2000 DATA 692,693,694,695,696,697,698,699,
2010 DATA 700,701,702,703,704,705,706,707,
2020 DATA 708,709,710,711,712,713,714,715,
2030 DATA 716,717,718,719,720,721,722,723,
2040 DATA 724,725,726,727,728,729,730,731,
2050 DATA 732,733,734,735,736,737,738,739,
2060 DATA 740,741,742,743,744,745,746,747,
2070 DATA 748,749,750,751,752,753,754,755,
2080 DATA 756,757,758,759,760,761,762,763,
2090 DATA 764,765,766,767,768,769,770,771,
2100 DATA 772,773,774,775,776,777,778,779,
2110 DATA 780,781,782,783,784,785,786,787,
2120 DATA 788,789,790,791,792,793,794,795,
2130 DATA 796,797,798,799,800,801,802,803,
2140 DATA 804,805,806,807,808,809,810,811,
2150 DATA 812,813,814,815,816,817,818,819,
2160 DATA 820,821,822,823,824,825,826,827,
2170 DATA 828,829,830,831,832,833,834,835,
2180 DATA 836,837,838,839,840,841,842,843,
2190 DATA 844,845,846,847,848,849,850,
```



# TROJAN CAD-MASTER

THE ULTIMATE IN GRAPHICS TOOLS

**SUPERB GRAPHICS SOFTWARE PLUS A TOP QUALITY LIGHT PEN**

Discover the exciting world of creating your own graphics on screen.

- **FREEDRAW-DRAW** - 5 pen thicknesses inc. Quills
- **PAINT BRUSH** - for the artistic touch
- **FILL ANY SHAPE** - use 16 colours and 11 patterns.
- **GEOMETRIC SHAPES** - circles, boxes, triangles, lines & scaling.
- **DYNAMIC RUBBER-BANDING** on all geometric options.
- **PIR-POINT FUNCTION** - for pixel accuracy on all functions.

Plus many more free functions to mention in this ad. All these features in the Program - a top-quality Light Pen and an instruction booklet in one reasonably priced package. Easy to use for creating colourful pictures or technical drawings. Ideal for use by all ages of micro-paint. Full back-up service from manufacturers. Available at good dealers or direct from Trojan Products.

Only £29.95 per pack



**TROJAN**  
Micro Computer Software & Accessories

Head Office: Trojan Products Ltd  
106, Denham, Uxbridge, Middlesex UB8 3PH  
Tel: 0753 506411  
TRADE ENQUIRIES WELCOME

## U.K. Soft Centre

COMMODORE SOFTWARE SPECIALISTS

Software Title	Price	Price
Amiga 500	£19.95	£19.95
Amiga 500 Plus	£29.95	£29.95
Amiga 500 Plus 2	£39.95	£39.95
Amiga 500 Plus 3	£49.95	£49.95
Amiga 500 Plus 4	£59.95	£59.95
Amiga 500 Plus 5	£69.95	£69.95
Amiga 500 Plus 6	£79.95	£79.95
Amiga 500 Plus 7	£89.95	£89.95
Amiga 500 Plus 8	£99.95	£99.95
Amiga 500 Plus 9	£109.95	£109.95
Amiga 500 Plus 10	£119.95	£119.95
Amiga 500 Plus 11	£129.95	£129.95
Amiga 500 Plus 12	£139.95	£139.95
Amiga 500 Plus 13	£149.95	£149.95
Amiga 500 Plus 14	£159.95	£159.95
Amiga 500 Plus 15	£169.95	£169.95
Amiga 500 Plus 16	£179.95	£179.95
Amiga 500 Plus 17	£189.95	£189.95
Amiga 500 Plus 18	£199.95	£199.95
Amiga 500 Plus 19	£209.95	£209.95
Amiga 500 Plus 20	£219.95	£219.95

Software Title	Price	Price
Amiga 500 Plus 21	£229.95	£229.95
Amiga 500 Plus 22	£239.95	£239.95
Amiga 500 Plus 23	£249.95	£249.95
Amiga 500 Plus 24	£259.95	£259.95
Amiga 500 Plus 25	£269.95	£269.95
Amiga 500 Plus 26	£279.95	£279.95
Amiga 500 Plus 27	£289.95	£289.95
Amiga 500 Plus 28	£299.95	£299.95
Amiga 500 Plus 29	£309.95	£309.95
Amiga 500 Plus 30	£319.95	£319.95
Amiga 500 Plus 31	£329.95	£329.95
Amiga 500 Plus 32	£339.95	£339.95
Amiga 500 Plus 33	£349.95	£349.95
Amiga 500 Plus 34	£359.95	£359.95
Amiga 500 Plus 35	£369.95	£369.95
Amiga 500 Plus 36	£379.95	£379.95
Amiga 500 Plus 37	£389.95	£389.95
Amiga 500 Plus 38	£399.95	£399.95
Amiga 500 Plus 39	£409.95	£409.95
Amiga 500 Plus 40	£419.95	£419.95

Software Title	Price	Price
Amiga 500 Plus 41	£429.95	£429.95
Amiga 500 Plus 42	£439.95	£439.95
Amiga 500 Plus 43	£449.95	£449.95
Amiga 500 Plus 44	£459.95	£459.95
Amiga 500 Plus 45	£469.95	£469.95
Amiga 500 Plus 46	£479.95	£479.95
Amiga 500 Plus 47	£489.95	£489.95
Amiga 500 Plus 48	£499.95	£499.95
Amiga 500 Plus 49	£509.95	£509.95
Amiga 500 Plus 50	£519.95	£519.95
Amiga 500 Plus 51	£529.95	£529.95
Amiga 500 Plus 52	£539.95	£539.95
Amiga 500 Plus 53	£549.95	£549.95
Amiga 500 Plus 54	£559.95	£559.95
Amiga 500 Plus 55	£569.95	£569.95
Amiga 500 Plus 56	£579.95	£579.95
Amiga 500 Plus 57	£589.95	£589.95
Amiga 500 Plus 58	£599.95	£599.95
Amiga 500 Plus 59	£609.95	£609.95
Amiga 500 Plus 60	£619.95	£619.95

Software Title	Price	Price
Amiga 500 Plus 61	£629.95	£629.95
Amiga 500 Plus 62	£639.95	£639.95
Amiga 500 Plus 63	£649.95	£649.95
Amiga 500 Plus 64	£659.95	£659.95
Amiga 500 Plus 65	£669.95	£669.95
Amiga 500 Plus 66	£679.95	£679.95
Amiga 500 Plus 67	£689.95	£689.95
Amiga 500 Plus 68	£699.95	£699.95
Amiga 500 Plus 69	£709.95	£709.95
Amiga 500 Plus 70	£719.95	£719.95
Amiga 500 Plus 71	£729.95	£729.95
Amiga 500 Plus 72	£739.95	£739.95
Amiga 500 Plus 73	£749.95	£749.95
Amiga 500 Plus 74	£759.95	£759.95
Amiga 500 Plus 75	£769.95	£769.95
Amiga 500 Plus 76	£779.95	£779.95
Amiga 500 Plus 77	£789.95	£789.95
Amiga 500 Plus 78	£799.95	£799.95
Amiga 500 Plus 79	£809.95	£809.95
Amiga 500 Plus 80	£819.95	£819.95

Software Title	Price	Price
Amiga 500 Plus 81	£829.95	£829.95
Amiga 500 Plus 82	£839.95	£839.95
Amiga 500 Plus 83	£849.95	£849.95
Amiga 500 Plus 84	£859.95	£859.95
Amiga 500 Plus 85	£869.95	£869.95
Amiga 500 Plus 86	£879.95	£879.95
Amiga 500 Plus 87	£889.95	£889.95
Amiga 500 Plus 88	£899.95	£899.95
Amiga 500 Plus 89	£909.95	£909.95
Amiga 500 Plus 90	£919.95	£919.95
Amiga 500 Plus 91	£929.95	£929.95
Amiga 500 Plus 92	£939.95	£939.95
Amiga 500 Plus 93	£949.95	£949.95
Amiga 500 Plus 94	£959.95	£959.95
Amiga 500 Plus 95	£969.95	£969.95
Amiga 500 Plus 96	£979.95	£979.95
Amiga 500 Plus 97	£989.95	£989.95
Amiga 500 Plus 98	£999.95	£999.95
Amiga 500 Plus 99	£1009.95	£1009.95
Amiga 500 Plus 100	£1019.95	£1019.95

Amiga 500 Plus 101 £1029.95  
Amiga 500 Plus 102 £1039.95  
Amiga 500 Plus 103 £1049.95  
Amiga 500 Plus 104 £1059.95  
Amiga 500 Plus 105 £1069.95  
Amiga 500 Plus 106 £1079.95  
Amiga 500 Plus 107 £1089.95  
Amiga 500 Plus 108 £1099.95  
Amiga 500 Plus 109 £1109.95  
Amiga 500 Plus 110 £1119.95  
Amiga 500 Plus 111 £1129.95  
Amiga 500 Plus 112 £1139.95  
Amiga 500 Plus 113 £1149.95  
Amiga 500 Plus 114 £1159.95  
Amiga 500 Plus 115 £1169.95  
Amiga 500 Plus 116 £1179.95  
Amiga 500 Plus 117 £1189.95  
Amiga 500 Plus 118 £1199.95  
Amiga 500 Plus 119 £1209.95  
Amiga 500 Plus 120 £1219.95

Amiga 500 Plus 121 £1229.95  
Amiga 500 Plus 122 £1239.95  
Amiga 500 Plus 123 £1249.95  
Amiga 500 Plus 124 £1259.95  
Amiga 500 Plus 125 £1269.95  
Amiga 500 Plus 126 £1279.95  
Amiga 500 Plus 127 £1289.95  
Amiga 500 Plus 128 £1299.95  
Amiga 500 Plus 129 £1309.95  
Amiga 500 Plus 130 £1319.95  
Amiga 500 Plus 131 £1329.95  
Amiga 500 Plus 132 £1339.95  
Amiga 500 Plus 133 £1349.95  
Amiga 500 Plus 134 £1359.95  
Amiga 500 Plus 135 £1369.95  
Amiga 500 Plus 136 £1379.95  
Amiga 500 Plus 137 £1389.95  
Amiga 500 Plus 138 £1399.95  
Amiga 500 Plus 139 £1409.95  
Amiga 500 Plus 140 £1419.95

Software Title	Price	Price
Amiga 500 Plus 141	£1429.95	£1429.95
Amiga 500 Plus 142	£1439.95	£1439.95
Amiga 500 Plus 143	£1449.95	£1449.95
Amiga 500 Plus 144	£1459.95	£1459.95
Amiga 500 Plus 145	£1469.95	£1469.95
Amiga 500 Plus 146	£1479.95	£1479.95
Amiga 500 Plus 147	£1489.95	£1489.95
Amiga 500 Plus 148	£1499.95	£1499.95
Amiga 500 Plus 149	£1509.95	£1509.95
Amiga 500 Plus 150	£1519.95	£1519.95
Amiga 500 Plus 151	£1529.95	£1529.95
Amiga 500 Plus 152	£1539.95	£1539.95
Amiga 500 Plus 153	£1549.95	£1549.95
Amiga 500 Plus 154	£1559.95	£1559.95
Amiga 500 Plus 155	£1569.95	£1569.95
Amiga 500 Plus 156	£1579.95	£1579.95
Amiga 500 Plus 157	£1589.95	£1589.95
Amiga 500 Plus 158	£1599.95	£1599.95
Amiga 500 Plus 159	£1609.95	£1609.95
Amiga 500 Plus 160	£1619.95	£1619.95
Amiga 500 Plus 161	£1629.95	£1629.95
Amiga 500 Plus 162	£1639.95	£1639.95
Amiga 500 Plus 163	£1649.95	£1649.95
Amiga 500 Plus 164	£1659.95	£1659.95
Amiga 500 Plus 165	£1669.95	£1669.95
Amiga 500 Plus 166	£1679.95	£1679.95
Amiga 500 Plus 167	£1689.95	£1689.95
Amiga 500 Plus 168	£1699.95	£1699.95
Amiga 500 Plus 169	£1709.95	£1709.95
Amiga 500 Plus 170	£1719.95	£1719.95

Software Title	Price	Price
Amiga 500 Plus 171	£1729.95	£1729.95
Amiga 500 Plus 172	£1739.95	£1739.95
Amiga 500 Plus 173	£1749.95	£1749.95
Amiga 500 Plus 174	£1759.95	£1759.95
Amiga 500 Plus 175	£1769.95	£1769.95
Amiga 500 Plus 176	£1779.95	£1779.95
Amiga 500 Plus 177	£1789.95	£1789.95
Amiga 500 Plus 178	£1799.95	£1799.95
Amiga 500 Plus 179	£1809.95	£1809.95
Amiga 500 Plus 180	£1819.95	£1819.95
Amiga 500 Plus 181	£1829.95	£1829.95
Amiga 500 Plus 182	£1839.95	£1839.95
Amiga 500 Plus 183	£1849.95	£1849.95
Amiga 500 Plus 184	£1859.95	£1859.95
Amiga 500 Plus 185	£1869.95	£1869.95
Amiga 500 Plus 186	£1879.95	£1879.95
Amiga 500 Plus 187	£1889.95	£1889.95
Amiga 500 Plus 188	£1899.95	£1899.95
Amiga 500 Plus 189	£1909.95	£1909.95
Amiga 500 Plus 190	£1919.95	£1919.95
Amiga 500 Plus 191	£1929.95	£1929.95
Amiga 500 Plus 192	£1939.95	£1939.95
Amiga 500 Plus 193	£1949.95	£1949.95
Amiga 500 Plus 194	£1959.95	£1959.95
Amiga 500 Plus 195	£1969.95	£1969.95
Amiga 500 Plus 196	£1979.95	£1979.95
Amiga 500 Plus 197	£1989.95	£1989.95
Amiga 500 Plus 198	£1999.95	£1999.95
Amiga 500 Plus 199	£2009.95	£2009.95
Amiga 500 Plus 200	£2019.95	£2019.95

Software Title	Price	Price
Amiga 500 Plus 201	£2029.95	£2029.95
Amiga 500 Plus 202	£2039.95	£2039.95
Amiga 500 Plus 203	£2049.95	£2049.95
Amiga 500 Plus 204	£2059.95	£2059.95
Amiga 500 Plus 205	£2069.95	£2069.95
Amiga 500 Plus 206	£2079.95	£2079.95
Amiga 500 Plus 207	£2089.95	£2089.95
Amiga 500 Plus 208	£2099.95	£2099.95
Amiga 500 Plus 209	£2109.95	£2109.95
Amiga 500 Plus 210	£2119.95	£2119.95
Amiga 500 Plus 211	£2129.95	£2129.95
Amiga 500 Plus 212	£2139.95	£2139.95
Amiga 500 Plus 213	£2149.95	£2149.95
Amiga 500 Plus 214	£2159.95	£2159.95
Amiga 500 Plus 215	£2169.95	£2169.95
Amiga 500 Plus 216	£2179.95	£2179.95
Amiga 500 Plus 217	£2189.95	£2189.95
Amiga 500 Plus 218	£2199.95	£2199.95
Amiga 500 Plus 219	£2209.95	£2209.95
Amiga 500 Plus 220	£2219.95	£2219.95
Amiga 500 Plus 221	£2229.95	£2229.95
Amiga 500 Plus 222	£2239.95	£2239.95
Amiga 500 Plus 223	£2249.95	£2249.95
Amiga 500 Plus 224	£2259.95	£2259.95
Amiga 500 Plus 225	£2269.95	£2269.95
Amiga 500 Plus 226	£2279.95	£2279.95
Amiga 500 Plus 227	£2289.95	£2289.95
Amiga 500 Plus 228	£2299.95	£2299.95
Amiga 500 Plus 229	£2309.95	£2309.95
Amiga 500 Plus 230	£2319.95	£2319.95

Software Title	Price	Price
Amiga 500 Plus 231	£2329.95	£2329.95
Amiga 500 Plus 232	£2339.95	£2339.95
Amiga 500 Plus 233	£2349.95	£2349.95
Amiga 500 Plus 234	£2359.95	£2359.95
Amiga 500 Plus 235	£2369.95	£2369.95
Amiga 500 Plus 236	£2379.95	£2379.95
Amiga 500 Plus 237	£2389.95	£2389.95
Amiga 500 Plus 238	£2399.95	£2399.95
Amiga 500 Plus 239	£2409.95	£2409.95
Amiga 500 Plus 240	£2419.95	£2419.95
Amiga 500 Plus 241	£2429.95	£2429.95
Amiga 500 Plus 242	£2439.95	£2439.95
Amiga 500 Plus 243	£2449.95	£2449.95
Amiga 500 Plus 244	£2459.95	£2459.95
Amiga 500 Plus 245	£2469.95	£2469.95
Amiga 500 Plus 246	£2479.95	£2479.95
Amiga 500 Plus 247	£2489.95	£2489.95
Amiga 500 Plus 248	£2499.95	£2499.95
Amiga 500 Plus 249	£2509.95	£2509.95
Amiga 500 Plus 250	£2519.95	£2519.95
Amiga 500 Plus 251	£2529.95	£2529.95
Amiga 500 Plus 252	£2539.95	£2539.95
Amiga 500 Plus 253	£2549.95	£2549.95
Amiga 500 Plus 254	£2559.95	£2559.95
Amiga 500 Plus 255	£2569.95	£2569.95
Amiga 500 Plus 256	£2579.95	£2579.95
Amiga 500 Plus 257	£2589.95	£2589.95

# Teacher's Pet

Margaret Webb goes back to school and learns about Basic.

**BEFORE LAUNCHING** FOURTH THIS month — a small confession. I've been writing this column for several months now and though my teaching qualifications are verifiable, the same cannot be said for my programming skills. I therefore decided that it was about time I rectified the situation by learning how to program in Basic. I was disappointed that this is not as easy as it would seem, the first question is to establish the best method of learning. Three approaches immediately spring to mind.

The first is to sign up at a local school or college for night classes. These are available for a number of languages with a bias towards BBC Basic. There are a number of traps however. Night classes are limited to the older age groups making them unavailable to youngsters. They also cost money.

If you want to use this route, you are constrained by a fixed timetable and term length. You must also learn when the teachers choose to teach. Some teachers have a strange idea as to what material a student needs. My sister is currently struggling through the mysteries of binary and hexadecimal as part of her course on Basic! If that was learning machine-code I could understand it, but not for Basic. It's a good idea to find a course which offers some practical work as well as lectures so that you can get hands-on experience with the help of the teacher.

A second approach is to find a home-based course and work through it under your own steam. The alternatives are either a correspondence course, if you can find one, or a book/software package. The main drawback with this method is that if you get stuck, you don't have anyone to bail you out.

Alternatively, you could find a tutor. There are always knowledgeable enthusiasts around who would spend some time passing on information for a small sum. The problem is finding them.

Probably the best approach for most people is to combine the latter two.

This month I intend to look at a number of book/software systems currently available.

Most of you will have discovered the lack of help provided when you buy a Commodore machine. The C-16 does come with a Basic tape but this is more of a sales gimmick, the actual manual are sent to order. This is often disastrous when you consider the tuition provided with other products. Many manufacturers of writing machines and musical instruments provide lessons or other teaching

material included in the price. Why can't computer manufacturers do it too!

One small word of warning. A large number of publishers have spotted the short comings of the Commodore manuals and have produced books teaching Basic and the inner workings of the computer. Many of these books are hardly any better than the Commodore manuals and you should purchase with care.

As a preliminary guide, I shall describe a number of packages covering different age-groups and prices.

Commodore offers a package for the Vic-20, C64, C-16 and Plus4 called *An Introduction to Basic*. Whilst originally sold as a separate package, it has subsequently been offered in the *Vicior Basic Packs*. You may be able to locate it separately if you look hard enough or contact Commodore.

The package comprises of two cassettes and a book. The manual is set out in a series of experiments which are linked to programs on tape. The material ranges from setting the computer up to fairly advanced programming. I was pleased to see the frequent use of flow diagrams to demonstrate the operation of the programs. The approach is structured so that you must cover the early material before progressing through the manual. Overall it's quite a reasonable system which does its job well enough.

For the younger users, Collins offers a book entitled *DATTA LOG*. This is a work book which has the appeal of a simple approach. Each page gently leads the user through the maze of programming, syntax and sound. The entire book is written in the form of a space ship's log with each entry or chapter adding to the contents, for example using screen printing and tabulation for the paragon for... *THAT'S* for rocket launch and using a spritz to design the captain's badge. The material is not covered in great depth but the book gives a good introduction to the use of Basic. At £1.95 it's also good value for money.

Glenzap Publishers have a wide range of teaching books and book/software packages available for a number of Commodore machines. Originally starting with the Dr Watson series of books for Basic and machine code, Glenzap moved on to the Watson's *Learn* series for the C64. This is a six book unit which deals with most facets of the C64.

The early volumes cover programming in Basic moving on to graphics in the later parts. The layout of the material is clear and tidy and is handled in a logical manner. If I do have a complaint it is that the content of the books is perhaps a little thin. At £1.95 per volume, I would have expected a little more depth. Notwithstanding this drawback, the series is well worth a close look.

Another good quality product from Glenzap is *Basic Adventure Part 1*. This book/software package teaches the rudiments of Basic to the more to 12 age group. The book takes the form of a science fiction story involving Dr Watson. Programming concepts are introduced in small routines which complement the text. Overall a novel and effective way of introducing teaching programming.

Commodore 64 Basic from Wiley is a really introductory to Basic. This is part of a Self Teaching guide series and uses self checking and exercises to aid you. The material covered is much as expected covering all facets of Basic, graphics and sound. However, are good extensively helping the pupil to develop a logical approach to programming. The book is sufficiently comprehensive that even once you have mastered programming in Basic, there are more advanced areas to look at. This is exemplified by the sections on databases and data storage on disk and cassette. The author is clearly American and this does unfortunately become apparent in the text with the introduction of *www* (American colloquialism). Provided you can stand the literary style, it's book is pretty good value albeit a little pricey.

For C-16 owners, Wile offers the *Gateway to Programming* series. This quartet of books tackles the subject in a similar manner to the Dr Watson series by using Sherlock Holmes stories to illustrate ideas. The text is humorous with lots of cartoon illustrations. In spite of the £4.95 price per volume, these are good value for money and worth a look.

Dawson Elderberry produces a Commodore 64 *Master Pack* consisting of books one and two of their *Screen shots* books and a typing tutor cassette. The screen shots are full colour pictures of things as they appear on the monitor and illustrations of the results you can expect from the program. The books step gently through learning Basic and the cassette should help you to get to grips with the keyboard. The *Screen shots* costs £11.95 but if the price is off-putting each book in the set can be bought separately.

All of the books described here have the virtue of using a well thought out approach to teaching Basic. In addition most are cheap and almost within pocket money range.



# INTERRUPTS

**LK Gibson introduces**

**C64 users to IRQ**

**interrupts.**

## What is Multi-Tasking?

THIS TERM IS USED TO DESCRIBE a computer that can run more than one program at any given time. Each program being transparent to the other. Output to the video monitor or TV screen is usually split so that windows are formed, each window being a miniature version of the normal screen for each program. This system of running programs concurrently can either be produced by hardware or software.

In the examples we are going to consider there are no windows and the ability to run two programs is derived purely from software.

Unfortunately this operation can only be performed in machine code, but don't despair. Basic programmers, the steps to have two programs up and running simultaneously are relatively simple and will be dealt with in as simple and concise a manner as possible.

## How it Works

The way in which we make a program run apparently transparent to anything that might be running is by making use of the system's IRQ INTERRUPTS. This interrupt is called 60 times every second (or once every 16 $\mu$ s).

When an IRQ occurs your C64 stops whatever it is doing, whether it's a Basic or a machine code program and goes off to carry out its own little machine code program. Remember, this happens so fast that it is transparent to the system. The machine code, executed during an IRQ, is simply a housekeeping routine and does things like SCAN THE KEYBOARD, UPDATE THE SYSTEM CLOCK, etc.

When an IRQ occurs your C64 must know where to find this block of housekeeping code. It finds an answer from memory locations 768 and 769 (HEX \$0034 & \$0035). The two numbers stored in these addresses form the INDIRECT starting address of the housekeeping code.

You may have noticed something funny. Memory locations 768 and 769 are situated in RAM and that means that the information stored there can be changed at will. To believe the processor tells the computer to go and do the housekeeping we redirect it to do whatever we want it to do. As you now know, IRQ occurs 60 times every second so we now have a program that runs 60 times every second irrespective of whatever else your computer may be doing at the time. This forms the basis for running at least two programs concurrently. Now we shall go on to consider in more technical terms exactly how this process is achieved.

As stated earlier, memory addresses 768 and 769 contain the INDIRECT address for the start of the normal INTERRUPT CODE. Location 768 contains the low part of the INDIRECT ADDRESS and 769 forms the high part of the INDIRECT ADDRESS. This method of calculating INDIRECT ADDRESS goes for any INDIRECT ADDRESS used by your computer, i.e. IOW RITE/WRIT RITE. Therefore, to find an actual address, we can use the formula:

$$\text{ADDRESS}(x) = \text{PIEC}(x) * 256$$

Where  $x$  is the first location (LOW BYTE) and  $ml$  is the second location (HIGH BYTE), i.e. 768 and 769.

Using this formula we can calculate the actual address of the standard INTERRUPT CODE with:

$$\text{ADDRESS}(768) + \text{PIEC}(769) * 256 = \text{PIECAD}$$

This will give the start address as 59951 (\$E437). Write down this address as we need to jump to it at the end of our own custom routine. If this is not done the system will not scan the keyboard and the result will be a system crash. Therefore the last instruction in our code must be:

**JMP \$E437**

All of this will become quite clear later so we don't worry if you're slightly confused at the moment. The programs given later are presented in such a way that they can be used by the novice computer owner, but will also form the basis of some more advanced INTER-

ruption in your program.

This code is fairly standard for setting up an interrupt driven software, although some programmers may wish to carry out some initialization for their program during this setting up procedure. The techniques for doing this will become apparent in later examples.

The routine as it stands will be situated in memory addresses \$F100 (HEX \$CB00), this is a 4K block of RAM situated above the Basic INTERPRETER. This is a convenient place away from the ranges of Basic, but almost any RAM location may be preferred provided you don't clash with Basic or the SYSTEM VARIABLES. Some

Address \$CB00	Opcodes \$C1	Comments
\$CB01	LDA #520	disable interrupts while setting up
\$CB05	STA \$E014	set low byte of start address for code store in a 768 decimal
\$CB06	LDA #608	set high byte of address for code store in a 768 decimal
\$CB08	STA \$E015	allow interrupts again
\$CB09	RTS	return to Basic
\$CB0A	JMP \$E437	examples given later jump to standard interrupt code

RIFT DRIVEN CODE for the more experienced amongst you.

An example of this is to form windows using RASTER INTERRUPTS effectively splitting the screen using one half for one program and the second half for your interrupt code, unfortunately this is beyond the scope of this article.

## How to Set up Interrupts

As an ASSEMBLY LANGUAGE program to set INTERRUPTS would go as follows:

**\$C000** is where your code begins and **JMP \$E437** is the last

useful locations are given below, although this list is by no means complete.

**\$C000(\$F100) to \$CFFF(\$FFFF)**. This is the space of RAM BLOCK situated above the Basic-INTERPRETER. As it is split from the normal Basic RAM (\$100-\$FFFF) it cannot be affected by Basic, therefore this is an ideal place for our INTERRUPT DRIVEN CODE.

Basic RAM \$0000(\$000) to \$FFFF(\$FFFF) Code can be placed at the top of Basic RAM but unless it is protected it will be overwritten by Basic STRING VARIABLES. To protect this area we must lower the top of Basic and also lower the bottom of STRING STORAGE. The pointers to

these are stored at 51-56 for TOP OF Basic and 51-52 for BOTTOM OF SCREEN ADDRESS. To protect the BASIC for our interrupt code we would use the program line:

```
10 POKE 51,110:52=1:POKE 54,POKE 56=1
```

To increase this to 512 BYTES of protected area we would substitute 2 for 1 in the above program line and so on for each 256 BYTES required.

Line 10 as it stands will give us 256 protected BYTES from location 40704 (\$F000) to 40959 (\$F03F) for our code.

**Cassette Buffer 428-429:** This is the cassette buffer and is totally safe for disk users but anything written there will be overwritten by cassette LOAD and SAVE operations, therefore cassette users must be very careful when placing code in this area.

From the above examples it can be seen that the nearest and most convenient addresses to place code are from \$F010 (\$C000) onwards, therefore all the examples given will use these addresses.

## Making It Work

Each example that follows will be preceded by a description of the program and will outline its purpose, this will then be followed by a Basic program containing the necessary code in data statements (this will allow users without a machine-code monitor to enter and run the programs given). Next will be given an assembly listing for those amongst you with machine code monitors and finally each example will be annotated to show you how it works.

### Fuzzy Border

The following program is the shortest example that I could think of. Although it doesn't really serve any practical purpose it does give dramatic examples of how INTERRUPT DRIVEN CODE works. When this program is completed you will see no difference to the 44 screen but the exterior border will be flicking dramatically, and will continue to do so even when you enter or load and run other programs.

Note that all these examples can be switched off by using the STOP/STOP and RETURN keys together.

### Basic Program 1

This is a complete Basic program and will automatically load the code when run. Type it in exactly as shown, save for security purposes and then run it and behold!

**Disassembled Listing 1 (with machine code monitor)**

```
$C000 SET
$C001 LDA #500
$C005 STA $B014
$C006 LDA #500
$C008 STA $B015
$C009 CLJ
$C00C RTS
$C020 LDX #600
$C022 STX $B020
$C025 DEX
$C028 BNE $C022
$C02B JMP $A011
```

If you try to enter Basic code once a machine code program has loaded the chances are you will get an "OUT OF MEMORY" error. There is a Basic BEEP program at the end of this article that will overcome this problem.

Hopefully you have now entered and fully understood Program 1, if this is not the case then I would strongly recommend that you go back and read the preceding paragraphs, which should be

disable interrupts, load acc with low byte of indirect address, store it in low byte of IRQ RAM vector, load acc with high byte of indirect address, store it in high byte of IRQ RAM vector, enable interrupts again, return to Basic.

set counter for screen colours, store it in border colour address, decrement colour counter (ie change the colour), is it done 100 times, if yes then jump to standard IRQ code before returning.

now had a lot more meaning. That introductory program wasn't too difficult was it, we'll move on to some more practical routines now, so sit down and prepare yourself to enter a fascinating sphere of computer programming.

### Using a Vic 1540 Disk Unit on the C64

If you've seen that bargain secondhand Vic 1540 Disk Drive and had to pass it up because you thought you couldn't use it on your C64 then this routine is for you. The 1540 disk drive will operate exactly the same as the 1541 when connected to a 64 apart from one major problem, it won't load programs without you first blinking on the screen and then getting it up again when loading is complete. This is a real nuisance and there are in fact two ways of overcoming this problem. 1 Buy a new 1541 ROM chip from Commodore for £24 and

try to fit it yourself, or pay a dealer to fit it at extra cost, or 2 Use this routine!

To use the program P1 blinks the screen and P1 again, therefore to load a program use the following procedure:

- 1 Type "LOAD" "YOUR PROG" ;
- 2 PRESS P1 and HIT RETURN
- 3 Once the red "disk in use" light has extinguished press P1

It's as simple as that!

By the way a more exotic way of overcoming the problem is to blank and open the screen automatically using the RAM LOAD and SAVE vectors, but again that is beyond the scope of this article, sorry.

### Basic Program 2

Remember to save program 2 before you execute it!

Once you have saved this program for getting it back in the computer and up and running, are exactly the same as that for PROGRAM 1.

If you're still with us and have at least partly understood the procedure involved for the programs above, then you should try now for starting to realise some of the weird and wonderful things that can be achieved with Interrupt Driven Code.

We now come to our final program concerning Interrupts and as you would expect it is also the most complex we have dealt with to date.

### Defined Function Keys

This program gives us defined function keys, the keys are defined as follows:

- F1=Change border colour. This will swap through each of the 15 possible colours individually.
- F2=Change screen colour. This will swap through each of the 15 possible screen colours individually.
- F3=Repeat keys toggle. This will toggle between all keys repeat and cursor keys only repeat.
- F4=Processor pause. This allows Basic programs to be stopped in mid run and allows for map debugging and also a faster frame facility. It also works with program listings to allow easier reading. (Please note this should also work with most

#### PROGRAM BASIC PROGRAM 1

```
1 REM BASIC PROGRAM 1
10 FOR A=0 TO 15:POKE A,
:POKE A+256,A:NEXT
20 FOR A=0 TO 15:POKE A,
:POKE A+256,A:NEXT
30 SET A=32
40 DATA 120,140,12,180,30,1,
50,192,14,31,1,80,74
60 DATA 162,1,192,12,200,202,
200,250,11,49,234,0
70 PRINT "CODE NOW ACTIVE!"
111
```

If you have machine code monitor then enter the above code as shown, again you must always SAVE code before executing it, this cannot be stressed enough. Switch your set off and then on again, type "LOAD" "YOUR PROGRAM NAME" ;1 and hit RETURN. Once the code has completed loading type SET A=32 to activate the code.







# L.A.R.C.A.D.I.A.

## ANCIPITAL



### Flippo flips over Ancipital and gets hammered at chess.

YOU KNOW WHAT? IT'S BEEN A HELL OF a month! I topped my high score on Ancipital (shock!) It's a good game, actually, finally found a good chess game that doesn't have the pants off me, and not only that I've got a stack of hints/tips that'll knock yer eyes off!

#### 'Clip It and See

OK, here goes. Ancipital is not really a new game, I admit that. But it's certainly one of Jeff's better ones. I got a bit tired of most of his other stuff, the early out'n'out shooty ups are fine for a few moments, but quickly pall in the end. The experimental Mama Mama makes you want hurt, and I find it more than a little bit frustrating. Having a Killfreak cowering all over the screen totally out of revision control. (I rage you should use the Forest — lol.) Very funny!

Where was I? Oh, yes. Ancipital, or just plain 'Cippy to aficionados, is a true original. It was gravity, a truly all the real objective, and a good dash on tell a alternative universe. I like that. I like games which have a background story are more fun to play. You know, a little something to read before you power up the game, to get you into the feel of the story. That's where this animal very highly in my estimations, and that's in the little book you get with the package. Come on, software house! You're here to entertain us, so how about it.

What's that? What's my hi-score on 'Cippy? Nah, I'm too modest. Right? C'mon! Well, alright, if you must know, it was 36,888. Pretty good, huh?

#### Chess Mate

I want tell you about this... Look, I've not really known for being a chess player. Well, actually, I'm rubbish, but I do keep playing it, despite humiliating defeats to

man and machine alike. I dunno, it's akin to the fatal fascination some folk have with cat chases. The best chess game I've stumbled over recently, in my glumcy for defeat in Coliseum Chess 2.0. It's so easy to play; simply moving your pieces using the cursor, rather than the old long winded KPS-64 kind of input you have to suffer in others I could mention. It's a delightful game, bringing back some of the peace and quiet of the real game. I hate the pants off me every time!

#### Hint Me Daddio! (Right to the Bar)

Here are a few tips for your waterpail. Hope are if you've heard any of these before.

Get lost stilly: This game drives me bonkers! I thought I'd won the last of this as my invention, but alas not the game is back with new rooms, new challenges, and now a PColl in sight! (Baw! I will can't leave it alone though. Shin up the rope in the Cold House and you get into the Server System, goody. Go all the way up the back stairs to Hammer Lane. (Tap Off) Planes, slip across to On the Road, Up On The Barbed-wire, and finally the Main Perform A Quickening. (OK, Matthew, finish, I know you read Perry French Brothers Comic! Climb up the rope and you'll find yourself in the Waterhouse. Get to the top, jump up, and you'll find yourself in the Bookie Room. Grab the gems at the top of the Bookie, and off you go, you're in the Space Station. Once on the Station, find your way to the Transporter, and you'll

find yet another new system of rooms. There's also a guard appearance of a room from Marty Mares... good grief!

Backfords Road - Boulderdash II: On the first screen you have to blast a hole through a wall by dropping a rock on a fairy. Then you must clear the earth under a wall, and then drop rocks onto it. The wall is Magic, and it will create a gem for every rock that passes through it. General tip: The antelope will create gems if they are contained by rocks. Try watching gems from the side of piles of rocks and examine the way the rocks fall. There's an interesting chat on how to get seemingly enclosed gems out by utilizing creative watching. (Sounds painful, but never mind!)

Made from Backford next time.

#### Raven from the Fax File

My current favourite games are Ancipital (planetsoft), for reasons previously specified; Bounty Bob Strikes Back (LJ Gold) still my better platform game, beating the pants off Bubs, any day; Hammer Cameo II (Eyes/CBA) if only for the teasing and the Aysling; Rescue on Tractor (planetsoft/Archives) one of the most state of the art arcade games in existence, and if Jeff makes it, then so do I; Ball Blazer (Luxafire) probably the most cheating 3D game out, fast and furious; Rupert and the Trogglers (Pete Quakshook Ashbitch!) the worst! And finally, Back'n'Roll (Archives), to my mind, the only new game from these boys worth a look.

#### That's Yer Lot!

OK, that's all in the Saga of Yellow Peril! He's a long Way Off and C'mon! He's Fast of Strange Fruit... we have time for this month. Next time world... well actually you'd better wait and see. You I always tell the home, and what makes do I get? None. So off you go and write me a letter about your high scores. Go on! And don't turn to the next page until you're done!! Harumph!



# £10,000

# HOLIDAY BONANZA



Malware, Service, Support, Training  
all from your local Commodore centre.

**WELCOME TO THE WORLD OF COMMODORE**

## COMMODORE 64

Commodore 64 computer  
with 640K RAM  
1MB RAM  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive

**£199.00**

## COMMODORE 6502

Commodore 6502  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive

**£199.00**

## Free Guide 6502-6508

A free guide to the following  
Commodore 6502  
Commodore 6508  
Commodore 6508  
Commodore 6508  
Commodore 6508  
Commodore 6508  
Commodore 6508  
Commodore 6508

**£199.00**

## Special Member Price

Commodore 64  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive

**£199.00**

## 6502 and 1570 Package

Commodore 6502  
1570 disk drive  
1570 disk drive  
1570 disk drive  
1570 disk drive  
1570 disk drive  
1570 disk drive  
1570 disk drive

**£199.00**

## Commodore 6502

Commodore 6502  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive

**£199.00**

**Yes its true !!!**

For a limited period we are giving away a **£5.00**  
Travel Cheque for spending money on your next  
holiday with **EVERY £50 SPENT**.

That's not all folks !!!

your vouchers could also win you a **FREE** stay  
**HOLIDAY** for two or four people in our lucky  
numbers draw.

## 6502

Commodore 6502  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive

**£199.00**

## 64 & 128

Commodore 64  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive

**£199.00**

## COMMODORE

Commodore 64  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive

**£199.00**

## Must be COMMODORE

Commodore 64  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive

**£199.00**

## CHROMASONIC

**Computer Centres**

48 Junction Road, Aylesbury, London N19 5RD  
01-263-9483/5  
238 Russell Hill Broadway, London NW10 2BA  
01-863-3704

## Special Member Price

Commodore 64  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive  
1541 disk drive  
1545 disk drive

**£199.00**



**"INSTANT CREDIT"**

**Terms**

**"INSTANT CREDIT"**

All products are guaranteed for one year unless otherwise stated. Payment may be made by Access, Barclaycard, Bankerscard, Building Society's cheque, cash or postal order. Sorry, cheques need five days for clearance. We reserve the right to change prices without prior notice. All prices are inclusive of VAT. Please check before ordering for carriage charges.

Stuart Cooke goes bonkers  
about Gremlin Graphics  
Bouncer.

# GAME

of the month



EVERY ONCE IN A WHILE A GAME arrives that you put in your disk drive, play for five minutes, think it isn't very good and put it away. Five minutes later you'll stick it back in the cassette recorder and suddenly find that it's there a'clock in the morning and you've been playing the game for hours. Well, Bouncer is one of those games.

Remember the game that you used to play when you were kids (or maybe you still do) where you can't step on any of the lines around the flagstones or the monster will get you? Well, Bouncer certainly owes a little of its background to this.

You play the part of a bouncing tennis ball travelling across a scrolling path. Your way is hindered by a large number of pitfalls. Perhaps the main problem that you have to overcome is the fact that you can only land on the paving stones. If you miss them your tennis ball plunges to the ground a long way below you. I'm not quite sure why the path is floating above ground, but you can't knock the programmer for a lack of artistic licence.

Large gaps in the paving can be bridged across by means of the super

bouncers. These are paving stones which have an arrow on them and give your ball that extra power it needs to bridge the gap.

Large walls block your path so you must guide your ball around them. In the meantime you can hanggle and float; logs are trying to stop your progress.

If (or when) you manage to reach the end of a level you'll find the goal awaiting your ball. If you enter this then you'll find your score increasing.

After each level you'll find yourself entering the bonus screen. This consists of a number of paving stones with question marks. Landing on a question mark increases your score. Be warned you only have a limited number of jumps in which to increase your score. Extremely lowly but it certainly bumps up your points.

Scattered around each of the levels you will find a number of myrror paving stones. These have the same mark as those on the bonus level but will not always give you more points. If you are lucky then you will gain extra jumps for use on the bonus level, extra points or extra balls. If however you are unlucky you will find

your ball being changed to death by a mouth or even punctured by a flying dart.

The hazards become more severe the further you get into the game. Mines are launched from the sea, rain (ball) traps suddenly appear and burst your ball, a plot hase of flying boats get in your way, even some of the paving stones disappear from beneath you. In fact it seems that everything is out to prevent you from reaching the goal and bonus level.

A superb tune sets the atmosphere for your track along the pathway. The graphics are excellent and your ball spins as though it has just left a tennis star's racquet.

As a bonus an extra game, Metaballs, is being given away free with Bouncer. Metaballs is a conversion of an earlier Spectrum game and offers very little difference from the original.

You play the part of a little bird who is flying around a vast number of locations attempting to stay away from the numerous hazards. Your aim in life is to collect a number of radioactive plants scattered around in some extremely inconvenient places.

Mapping the locations for this game is definitely a must as you will soon find yourself lost.

Metaballs offers nothing out of the ordinary and probably wouldn't do too well as a stand alone game. However being included in a package with Bouncer makes it an excellent purchase.

Bouncer is definitely one of those 'just another 100' type of games and a must for any serious Commodore 64 collection. Especially when you remember that you are getting another game thrown in for the price of one.

# OR... MADE BY ALIEN



## THE FINAL CARTRIDGE®

THE FIRST OUTSIDE OPERATING SYSTEM FOR THE CBM 64

This new operating system built in a cartridge does not use any memory and is always there. Compatible with 99% of all programs.

### Features:

• **DESK TURBO** - 4 times faster disk access, loading and saving.

• **TAPE TURBO** - 10 times faster, even with files - normal Commodore commands - compatible with standard tape's.

• **ADVANCED CARTRIDGE INTER-FACE** - compatible with all the well-known cartridge printers and Commodore's printer programs. Prints all the Commodore graphics and control codes (important for coding).

Advanced system dump facilities. Prints Low Res, Hi-Res and Multires. Four FullpagePrinters from games and other programs. Map, Double, Border, Pad etc. Screen's compatible for the memory address of the picture.

• **THE EXTRA BONUS FOR BASIC PROGRAMS AVAILABLE** They now use words "Memory read". "Memory on". They move 192 bytes with double-intelligence speed, available in the 64K Ram of the CBM 64. Can be used with keys and variables.

• **BASIC 4 & 5 COMMANDS** (Go, Draw, Gaps, Catalog, etc.)

• **BASIC TOOLKIT** with Auto, Run, Quit, Save and Delete's Find, Help, Del, etc.

• **PREPROGRAMMED FUNCTION KEYPAD** - Run, Load, Save, Catalog, Quit commands. List preserves all list instructions.

• **KEYBOARD EXTRA'S** - Allows you to delete part of a line, stop and copy some strings, reset cursor to laster leftmost column. Home and Beyond in File. Top command operates your pointer as a typewriter.

• **COMFORTABLE EXTENDED ML MODE EDITOR** - with expanded load scrolling up and down, bank switching, etc.

• **RESET SWITCH** - resets to monitor, resets with end, resets to the Run position, resets every protected program.

• **ON/OFF SWITCH** - we hope you never need that one.

12 Months triple warranty guarantee.

12 (plus monthly) back guarantee if you are dissatisfied.

**H & P**  
4 0344 4112 1411

**U.K. ORDERS** Bentley's and Associates Ltd. Cheques should be made out to: H & P Computers 17 C/O Haverhill Wokingham Road CM9 3 5Z England. Telephone: 0378 - 511471.

Copyright not registered worldwide. All rights reserved. Reproduction in any form without permission is prohibited.

### SPECIAL INTRODUCTORY PRICE

FOR ONE € 50,- FOR TWO € 37,- each

FOR THREE OR MORE € 30,- each

Don't wait for your friends. ORDER ONE NOW!!! Just pay the difference if you reorder within a month.





Lines 3000 to 3710 reduce the number of columns of characters on the screen to 80 by blanking out the first and last. This means that characters will smoothly scroll off the edges of the screen; again, try running there and watch the left hand side.

The end of this routine simply places the data in lines 878 and 880 on to the screen, from the 'AND' statements. There are used because the assembler converts alpha-numerics in BTL statements into their ASCII values. The C4 screen, however, uses values 64 less than the ASCII equivalent. These 'AND's, therefore, convert bit seven and six from the values - (the equivalent of '64').

The routine "HAROLD", is designed to be totally portable, that is, you can use it in any program. Five tables are used: 4500 SCROLL : this contains the 4 screen scroll values. 4500 RAST2 : the Y positions where raster interrupts occur. 4500 BORDER : the border colors.

1400 UNIVERSITY (the address of the small station).

I have allowed six interrupt positions - you can add more or use less - and it is easily possible to add more tables defining further effects; I usually would use instance.

Now we know that if any interrupt occurs the processor will complete what it is doing and jump to **HANDLER**. The first thing we must decide is whether the interrupt has been caused by our timer compare value or by some other source. This is done in lines 390 to 450. When a timer interrupt occurs, bit zero of **INT0** is set to one. If this is the case we jump to **HAIR** - the normal SERIAL routine. A 2 must be written back into **INT0** to clear the register, ready for the next interrupt (a rather strange way to do it if you ask me!) in line 460.

The next column line checks the value in **IBB3** (the current raster position) against our table of values, and branches to

4150	JMP AMYPO2		
4151	;		
4152	LSI #0		
4153	JMP AMYPO2		
4154	;		
4155	LSI #0		
4156	JMP AMYPO2		
4157	;		
4158	LSI #4	4170	BACK1
4159	JMP AMYPO2	4180	;
4160	;	4190	;
4161	LSI #0	41A0	MOV1
4162	;	41B0	;
4163	LSA #0014	41C0	MOV2
4164	AND #040	41D0	;
4165	ORA SCROLL1,1	41E0	;
4166	STA #0014	41F0	ORA SCROLL1+1
4167	LSA #0011,1	4100	ORA SCROLL1+1
4168	STA #0011	4110	ORA SCROLL1+1
4169	LSA #0011,1	4120	STA SCROLL1+1
4170	STA #0011	4130	STA SCROLL1+1
4171	LSA #0011,1	4140	STA SCROLL1+1
4172	STA #0011	4150	STA SCROLL1+1
4173	LSA #0011,1	4160	STA SCROLL1+1
4174	STA #0011	4170	STA SCROLL1+1
4175	LSA #0011,1	4180	STA SCROLL1+1
4176	STA #0011	4190	STA SCROLL1+1
4177	LSA #0011,1	41A0	STA SCROLL1+1
4178	STA #0011	41B0	STA SCROLL1+1
4179	LSA #0011,1	41C0	STA SCROLL1+1
4180	STA #0011	41D0	STA SCROLL1+1
4181	LSA #0011,1	41E0	STA SCROLL1+1
4182	STA #0011	41F0	STA SCROLL1+1
4183	LSA #0011,1	4100	STA SCROLL1+1
4184	STA #0011	4110	STA SCROLL1+1
4185	LSA #0011,1	4120	STA SCROLL1+1
4186	STA #0011	4130	STA SCROLL1+1
4187	LSA #0011,1	4140	STA SCROLL1+1
4188	STA #0011	4150	STA SCROLL1+1
4189	LSA #0011,1	4160	STA SCROLL1+1
4190	STA #0011	4170	STA SCROLL1+1
4191	LSA #0011,1	4180	STA SCROLL1+1
4192	STA #0011	4190	STA SCROLL1+1
4193	LSA #0011,1	41A0	STA SCROLL1+1
4194	STA #0011	41B0	STA SCROLL1+1
4195	LSA #0011,1	41C0	STA SCROLL1+1
4196	STA #0011	41D0	STA SCROLL1+1
4197	LSA #0011,1	41E0	STA SCROLL1+1
4198	STA #0011	41F0	STA SCROLL1+1
4199	LSA #0011,1	4100	STA SCROLL1+1
4200	STA #0011	4110	STA SCROLL1+1
4201	LSA #0011,1	4120	STA SCROLL1+1
4202	STA #0011	4130	STA SCROLL1+1
4203	LSA #0011,1	4140	STA SCROLL1+1
4204	STA #0011	4150	STA SCROLL1+1
4205	LSA #0011,1	4160	STA SCROLL1+1
4206	STA #0011	4170	STA SCROLL1+1
4207	LSA #0011,1	4180	STA SCROLL1+1
4208	STA #0011	4190	STA SCROLL1+1
4209	LSA #0011,1	41A0	STA SCROLL1+1
4210	STA #0011	41B0	STA SCROLL1+1
4211	LSA #0011,1	41C0	STA SCROLL1+1
4212	STA #0011	41D0	STA SCROLL1+1
4213	LSA #0011,1	41E0	STA SCROLL1+1
4214	STA #0011	41F0	STA SCROLL1+1
4215	LSA #0011,1	4100	STA SCROLL1+1
4216	STA #0011	4110	STA SCROLL1+1
4217	LSA #0011,1	4120	STA SCROLL1+1
4218	STA #0011	4130	STA SCROLL1+1
4219	LSA #0011,1	4140	STA SCROLL1+1
4220	STA #0011	4150	STA SCROLL1+1
4221	LSA #0011,1	4160	STA SCROLL1+1
4222	STA #0011	4170	STA SCROLL1+1
4223	LSA #0011,1	4180	STA SCROLL1+1
4224	STA #0011	4190	STA SCROLL1+1
4225	LSA #0011,1	41A0	STA SCROLL1+1
4226	STA #0011	41B0	STA SCROLL1+1
4227	LSA #0011,1	41C0	STA SCROLL1+1
4228	STA #0011	41D0	STA SCROLL1+1
4229	LSA #0011,1	41E0	STA SCROLL1+1
4230	STA #0011	41F0	STA SCROLL1+1
4231	LSA #0011,1	4100	STA SCROLL1+1
4232	STA #0011	4110	STA SCROLL1+1
4233	LSA #0011,1	4120	STA SCROLL1+1
4234	STA #0011	4130	STA SCROLL1+1
4235	LSA #0011,1	4140	STA SCROLL1+1
4236	STA #0011	4150	STA SCROLL1+1
4237	LSA #0011,1	4160	STA SCROLL1+1
4238	STA #0011	4170	STA SCROLL1+1
4239	LSA #0011,1	4180	STA SCROLL1+1
4240	STA #		

4870	STA 80407, Y	5090	LPH 0
4880	LDA 80800, Y	5100	LPH 004
4890	STA 80807, Y	5110	LOOP**
4900	JMP	5120	LPH 004A, Y
4910	CPY 80420	5130	STA 1045, Y
4920	BNE LOOP7	5140	JMP
4930	JSR 8040FILL	5150	CPY 8121
4940	JSR	5160	BNE LOOP8
4950	I	5170	LPH 1104+79
4960	MOVZ	5180	STY 1104+79
4970	JSR 8040FILL	5190	LPH 1044+79
4980	JSR	5200	STY 1044+79
4990	I	5210	STY 1044+79
5000	MOVZ	5220	JSR
5010	BNE 004	5230	STY
5020	LPH CLIPED	5240	I
5030	STX COUNT	5250	PAUSE
5040	BNE SCROLL1+5	5260	8040FILL
5050	LPH SCROLL1+5	5270	STX 80420+8C7
5060	BPL 004	5280	STY
5070	LDA 87	5290	CPY 70415H
5080	STX SCROLL1+5	5300	JMP



the appropriate "POSITION". At this point the value of X is set up ready to index into the tables.

The routine ANYPOS is used for all positions, and sets up the X scratch scroll, and interrupt position, background

and border colours and the address of the movement routine. The effect of this is to allow "bands" of characters on the screen to scroll at different speeds, in different directions and with different colours. No mean feat!

"PAUSE" simply restores the registers to their current values and returns from the interrupt.

### The Scroll Routines

Essentially, all the scroll routines from line 4020 to 5340 are the same, so I will concentrate on just one of them: "MOVZ".

The "band" in area four (the road) will be moving from right to left at a rate of two pixels every interrupt (8000 of a second). This is achieved in lines 4750 to 4860. If it were to move right we would increment the value in "SCROLL1+5".

Next we check to see if we have scrolled a whole character - lines 4880 and 4920. If the value has reached minus one we replace it with seven (lines 4930 and 4940) and proceed to scroll the character eight pixels (one character) to the left - lines 4950 to 4990.

"SCROLL1" is the routine which fills in the right-most character, in this case with the character which "falls" off the left.

It will be seen that "ANYPOS" and "PAUSE" have now been moved to the interrupt routine. This is to ensure that there is no flicker when the sprites are moved, since they are printed when the raster is below them.

### There's More

When typing in the listing, change the symbols "&" and ">" to "<" and ">" for example:

```
LDA R 8040D0 & 20 becomes
LDA R < 8040D0
```

This is because my assembler insists on being awkward when it does a listing! "START" should now be at location 8210 (8010) "THROW" should now be at location 8230 (8030).

Next month we shall introduce the frog movement routines, and a joystick reading routine.

17



## BINDERS

FOR YOUR VALUABLE  
COLLECTION OF  
YOUR COMMODORE  
MAGAZINES

"BRIGHT" "EASY" TO USE  
"TOP QUALITY"

£1.00  
per  
mag.

To: ASP Systems Services, PO Box 26, Welwyn  
Hemel Hempstead, Herts. HP1 4SD (0445-470370)

Please specify: Four Commodore Magazines £4.00 per set. ASP  
Mag 1, 2, 3, 4 (please state magazine number to ASP) and 5  
please specify: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

Name

Address  
Please allow 10 days for delivery

If an advertisement  
is wrong we're here  
to put it right.

If you see an advertisement in the press, in print,  
on posters or in the cinema which you find  
unsatisfactory, write to us at the address below.

The Advertising Standards Authority

ASA Ltd, Day's House, Trenchard Place, London WC2E 7TH

# THE • BEAT • GOES • ON •

**Syntron's Digidrum is foot-tapping, hand-clapping good, according to Eric Doyle.**

THE HEART OF A ROCKED BAND IS THE ability of the rhythm section to mark time with the accuracy of a metronome and, it is not surprising, that microchip technology has developed the rhythm generator as the ultimate musical tempo. The problem is that, unless you are willing to pay a small fortune, the current generation of machines has as much soul as the ticking of a clock. But this appears to be changing.

Syntron's Digidrum is a flexible drumkit synthesizer which allows for those little touches of individuality which normally differentiate the human from the machine. There is sufficient programming flexibility to allow changes in rhythm and variation in style and to convince the average listener that he is listening to a syndrum kit played by a human.

The Digidrum package consists of a set of chips, containing the computer software and instrument databases, and a cartridge which plugs into the user port of the C-M4. The cartridge has a standard jack output for connection to an external amplifier and a trigger output which can be used to keep several slave sequenced synthesizers in time to the beat.

On loading the software, you already have a standard drum kit of seven instruments: base drum, snaredrum, three tom toms (large, small and floor), a crash cymbal and a hi-hat. These give a range of eight sounds in all because the hi-hat cymbal is closed as two separate instruments whether open or closed. The sample programs give an opportunity to hear how the drumkit sounds and very impressive it is too, despite the slightly electronic sound.

The computer program is in two sections. The first part allows you to compose short drum patterns and the second permits you to combine these patterns into complete backing tracks. There is no facility for printing out either the patterns or the songs so I found the best method was to switch from one section to the next and assemble the song from each pattern whilst the sound was fresh in my mind.

The pattern composer will permit 51 different patterns with a length of 38 beats. This is an arbitrary figure because the tempo can be changed over 64 steps which gives a range from the very, very slow to the impossibly fast. The instructions suggest that the mean value is around 44. The screen display looks like a familiar musical staff but there are eight lines instead of the normal five, each with a letter corresponding to one of the instruments.

Notes are entered by moving the arrow cursor along a horizontal vertical scale at the bottom of the screen until the correct position is reached. Then the letter key corresponding to the chosen instrument is pressed. The result is that a symbol appears on the relevant line and the instrument sound is heard. After a few notes have been entered, you can listen to the pattern singly by pressing a function key.

The number of instruments which can be sounded at the same beat position is limited to three. A drummer only has two hands so the program makes it impossible for a snaredrum, tom tom and cymbal to be sounded at the same time but a snaredrum, cymbal and base drum can. The flexibility of this system means that a respectable and plausible drum solo can be generated using trios, conversations or any other device or basic rhythm which takes your fancy.

Each pattern does not have to be a full 38 beats long it can be terminated by placing an end bar at any position along its length.

Once a few patterns have been created they can be combined into a song using the second program. This is extremely simple to understand. The screen shows several columns which are tabulated in rows from one to 100. This is the maximum number of steps which a song can have but since each step can consist of the same pattern repeated 100 times it doesn't take an lifetime to work out that there is room enough for even the most ambitious project.

Entry of a song is made by selecting the pattern number and the number of repeats which are then displayed on the song screen. The song can then be played in full or part to see how it sounds and if necessary a pattern can be called up and trimmed until the fusion is complete.

As in a word processor, there are several keys which allow the selection of blocks of the song which can be copied, deleted or inserted. Similarly patterns can be copied from one pattern position to another so that small changes can be made to create a new variation to add interest to the generated rhythm.

There is room for 10 songs which use the same bank of patterns and these can be saved to disk for recall when necessary.

In addition to the basic drum kit there is the option to replace any or all of the instruments with new ones which range from the hi-hat syndrum sounds to the more unusual percussion instruments such as a metal bar or even a very realistic hand clap. The limitation is still eight sounds and only three to a beat.

In the studio it would be ideal for making demo tapes and the only complaint I have is the length of time taken to create the patterns. I did find this stage quite enjoyable, however, giving plenty of freedom to experiment. For live performance, Digidrum would only be practical if all the music used the same percussion set or the performance was organised to allow time for loading.

Although there are limitations to live performances, I am quite sure that it won't be long before the strains of Sid Syntron's scintillating syncopation are heard in the local pub.

Ian Waugh has been  
discovering Island Logic's The  
Music System.

THERE ARE ALREADY DOZENS OF music packages available for the C64, all with their various strengths and weaknesses. Anyone who launches yet another package must think they have something pretty special.

Even if your ears turn blue at the mere mention of a BBC computer, you may well have heard about Island Logic's The Music System which was hailed by music-minded users as the best thing since memory expansion boards. The development team, System Software, has now produced a version of the Music System for the C64 and 128 and distribution has switched to Finland.

As you might expect, the superiority of the MD chip immediately gives any Commodore music program a tremendous advantage over a similar one on the Beek. Not content with this, System Software has included a MIDI module which allows access to external synthesizers.

The Music System (or TMS as it is usually referred to) was highly regarded not only for its music features but also for its use of icons and pull-down menus. The Commodore version can only enhance System's programming and design reputation as its use of these features are not only superbly implemented but they also make its operation relatively simple. And with its modules to choose from and over a hundred functions available from the keyboard, it needs to be simple. Most keys perform the same functions in each module so it's nowhere as near so daunting a task as it may at first appear. A handy Quick Key Guide helps immensely and you'll find after a little use that the keys fall under your fingers quite naturally.

Right! Let's take a look at the modules. There are the Editor, Keyboard, Synthesizer, MIDI, Printer and Linker whose icons are displayed on the main menu screen. Each module has a Command Line running across the top of the screen from which the pull-down windows...er...pull down. There are four menu bars: Files, Values, Commands and Info and each is selected by pressing one of the function keys. The information given in the Command Line menu differs from module to module but is similar in type.

Files controls the loading, saving, renaming and deletion of files and only those relevant to the module you are in can be accessed from that module. Values holds such information as key signature, tempo, octave, volume and voice



number. Commands is generally concerned with instructions which affect the whole or large portions of the piece such as delete track and clear all tracks. It is also home for a set of macro commands such as setting markers, copying sections to the notepad (more of that in a moment), defining barlines, snapping and copying envelopes and filters, etc. Info displays general information about the state of your composition, for example note storage space and the names of current music and sound files.

Moving on to specific modules, the one you are likely to use the most is the Editor. This displays a treble and bass clef in what is referred to as the Voice Monitor (VMW). Notes are entered here. The VMW only shows one voice at a time but you can flip from one to another at the press of a key and the bars are always aligned.

Notes are moved up and down the staves to select pitch and the note name and octave is displayed in a small box on the top right of the screen. Each note can be assigned a different volume level and any one of 16 envelopes. The current bar number is shown and horizontal bars called barometers indicate how much has been recorded on each voice. You can scroll through the score with ease and notes can be inserted and deleted at any point. A full range of accidentals can be used including double sharps and flats for the musical intelligentsia and notes can be turned into triplets and tied (although no

more than two notes can be tied together at once).

The program will insert bar lines automatically if required and you can insert first and second time bars. Another feature of TMS is the ability to define loop sections. When played as part of a tune, a loop section will loop repeating until the whole tune has finished. Each voice can contain up to 30 different loop sections so you can quickly select one of a number of repeating bass or rhythm patterns to sequence over or for use in a tune.

If all these features leave your mouth watering and fingers itching it's only fair to warn you that we're only up to page 25 of the manual. There is lots more to come.

From the Commands menu you can call up a set of macro commands. These operate upon a section of a voice which has previously been marked with two markers. Macro commands include transposition and envelope and volume assignment.

Yet another feature is the Notepad. This is used to store a section of a voice which can be moved to another part of the same voice or a different one. It can also be used to merge two music files together and notepad files can be saved and loaded like any other although only one can be held in memory at a time.

If you prefer to tap out tunes on the Commodore's keys, enter the Keyboard module. Real-time note entry from QWERTY keys is not the easiest way of writing a tune but the Keyboard module

N  
O  
W



# HEAR



# THIS



helps to make a difficult job as easy as it can be. Three voices are entered monophonically with horizontal bar-meters to show you how much room is available for each voice and the TMS can be called to display the notes. The keyboard is entered in practice mode, or Tink as it is called in the manual. Music and sound files can be loaded although only music files can be saved from here. The manual is kind enough to explain that anything but simple tunes may produce untidy compositions – and it's right – but then the Editor module includes a Block Edit command to help put right the mess you make.

The Synthesizer module is where you come to grips with MD. It must be the most complete and sophisticated MD chip editor yet devised. Graphic displays give a visual indication of MD's parameters and you can load and play a music file to hear the effect of the envelope as you alter it on a proper tune. The envelope includes sustain and filter sections and you can name each creation for easy reference.

The Synthesizer includes extra facilities like sweeping the pulse width, pitch and filter without using another voice – an extra bit of magic.

The MDN module will be attractive to a lot of users but let us not forget that a MIDI interface is also required. The program supports the IBM (178) and the Busport (178) interfaces and although it may work with others this is not guaranteed. Although the concept of

MDN was to produce a standard set of digital information signals, the standards haven't yet filtered down to MIDI interface manufacturers.

This module is basically a six-track polyphonic real-time sequencer. It records most performance information but not pitch-bend which cuts off any following data. I wonder how this got past the debugging team. You can set the tempo, select and delete individual tracks and pause recording by pressing the space bar. There are no channel assignment or editing facilities but it is a rather excellent extra and TMS is not, after all, a dedicated MIDI program. A big plus is the ability to convert MIDI files to music files playable by MD and vice versa. Regrettable but rather obviously, any multi-part polyphonic pieces are converted into three monophonic lines. Performance data goes but voice numbers in the range one to 32 are converted to envelope numbers. You can use the MDN module for real-time input and tidy up the parts later in the Editor.

The Printer module supports Epson and Commodore printers and you can add bytes to the score, too, providing a convenient copy of your masterpiece.

Finally, the Linker module is used to link individual music files. It is the only way tempo, key and time signature changes can be implemented in a single piece and, of course, it allows numerous compositions to play through in total. Up to 26 files can be loaded, memory

permitting, and arranged to play back in a sequence which can contain up to 99 items. The whole arrangement can then be saved as one file for convenient reloading and playing although you can't play this back through the MDN module.

The 34 page manual is well-produced, well-written, easy to read, full of illustrations and a comprehensive index will lead you to virtually every occurrence of every aspect of TMS. Unlabeled demo tunes are supplied on the disk with even more on the other side (a bit naughty, these reversible disks, aren't they?).

You may have guessed by now, but if you haven't here it is: TMS is the ultimate Commodore music editor for the MD chip. The MDN module is a bonus although musicians with serious MIDI requirements will need a dedicated software package. TMS is easy to use in spite of its wealth of features and it's fun. If you are at all interested in making music with your Commodore, I can not recommend it too highly.

The Advanced version of TMS containing all the modules described above retails for £29.95 and is available only on disk. A smaller version containing only the Editor, Keyboard and Synthesizer modules sells for £17.95 on disk and £14.95 on cassette. What will system producer for the Amiga!

The Music System is available from: Hired Software, Wellington House, Upper St Martin's Lane, London, WC2H 9BS.

## Yak's Progress

Hardware £11.99 disk £9.95 cassette £6.95 joystick



JEFF MINTER HAS BEEN around since Vic was just a lad and now a collection of Minter games is available under the title of Yak's progress.

As a document of one man's

fight to tame a machine, this compendium is fascinating and allows the newer C64-owners to catch up on some of the best and most unusual zapping games to be devised for the machine.

Eight games for the price of one is an offer that few will refuse and for many it will provide the chance to get turbo versions of games already in their collections as well as filling up any gaps.

Attack of the Mutant Camels and Revenge are both here alongside Matrix, Laser Zone, Slings in Space, Megagalactic Mania, Antipodal and Flower Berries.

There are enough camels to give you the hump, sheep to drive you bananas and bananas to drive you bananas. The graphics vary as Minter's programming skills develop and the accompanying booklet gives the lowdown on the workings of the programmer's fevered brain.

To try to describe the games would probably take most of the review space in this issue because the games are unlike anything to be seen elsewhere. Imaginative and demanding, they are not everyone's cup of tea but the Minter following is by no means a minority movement.

**L.D.**

## The Last VII

Mastertronic Mad series £2.99 C64



"WE RETURN TO BASE immediately" crackles the voice in your headset and you're off in a race against time.

As you try to get home before a delayed attack nuclear warhead wipes you off the face of the earth.

As a scientist working on project VII, you were hidden deep underground at the time of the holocaust. You have now been allowed to try and make contact with the survivors.

The screen is divided into two windows. The bottom depicts your instrument panel whilst the top gives a bird's-eye view of you and your surroundings. The dash you have to watch has to be simple—they include speed, distance to base and time before the explosion. You will, however, have to keep your eyes firmly on the road.

Providing you stay on the road, all well and good but you

soon discover that you cannot take corners at 400kph. One crash and that's it—game over.

If you slow down to take corners, there is no time for you to return to base before the bomb detonates. Eventually, I realised that some of the trees I kept hitting could, with slightly better driving, be avoided.

Strangely, the speech adds little to the game's atmosphere but the rest is extremely addictive. The car is very responsive and the music is great. You don't have to be a racing fan to enjoy this one.

**G.H.**

## Blackwyche

Ultimate £9.95 C64 • joystick



MUCH HAS ALREADY BEEN said about the C64 answer to the Master Colours, the best ship Blackwyche. In my opinion, a

lot of this outgore hype. It is an adventure style game and has little to commend it over many similar offerings.

Starting on the upper decks you are immediately harried by winged demons and the occasional flying octopus! Unarmed and helpless, your first task is to examine the crates to find a sword guarded by two nasty skeletons. Attacking from both sides, they beat you with bony fingers, knocking down your energy level as they do so. You must find antidotes to regain your power.

Once you have the sword you can attack the flying monsters outside but it is ineffective against most of the ghouls in the other cabins.

Occasionally you will get a surprise as you enter a cabin, the floor gives way and you end up dazed on a lower deck. Ladders lead up and in this way you can tour the whole ship making discoveries but no friends.

The locations are numerous but the graphics are very repetitive. Only colours differentiate which deck you are on. There is some suitably nautical music over the opening screen but during play there are only the sound effects which are fairly unimaginative.

**L.D.**

## Dynamite Dan

Microsoft £19.95 C&A

5
 7
 7
 6



ARRIVING BY ZEPPELIN at the hideout of Doctor Blitzen is not the subtlest approach. Dynamite Dan could have

made. With the assistance of his henchwoman, Donna, the doctor has set into operation a multitude of defence systems

and it is up to nimble-footed Dan to avoid these obstacles, steal the plans for the super psychotronic mega-ray which Donna and Blitzen have secreted in the safe, and make his escape.

This is a new platform game from Microsoft and it is decidedly difficult. As you guide Dan around the house looking for the right sticks of dynamite needed to blow the safe, all manner of creatures have to be avoided while keeping an eye on your energy level. Food is found at regular intervals which will boost both Dan's energy and your score depending on the type of food found.

There are other objects which score points, but the most valuable are the tool tubes. The score for this discovery is a miserable 20 points but they do give Dan an extra life which is absolutely essential for success.

At the bottom of the house is a river and Dan displays what is quaintly referred to as 'negative buoyancy' - he can't swim. This is the most dangerous of all the game's elements because one dip in the water and all of Dan's lives are lost.

The game will challenge any platform fanatic with a yearning for explosive fun.

LD.

## Hero of the Golden Talamon

Mastertronic Mail Range £2.99 C&A

6
 4
 4
 5



MINING MY FIRST JUMP in Golden Talamon, I was more than a little surprised to find myself disorientated under

water being chased by a shoal of ferocious looking piranha. As usual, I had not read the instructions, otherwise I would

have realised that a fair proportion of this arcade adventure is of a subaquatic nature.

In order to find the five pieces of the talisman, you must manipulate candles, keys, ropes and spells as you make your way through the various tunnels. Assorted monsters including particularly vicious fire-breathing dragons are intent on stopping you. You can carry up to five items at once, the left-hand box showing the one currently in use. Also shown on the screen is a stylised map of where you have been, indicators of your strength remaining, oxygen carried and a large map of

immediate surroundings. Your strength and oxygen supplies can be increased by collecting fruit and bags of air that appear periodically.

Movement is straightforward, the only tricky bits being the jumps which involve the diagonals on the joystick. There is no scrolling between them so you can never see what you will find on the next one. The graphics are large and blocky with some of the movement being jerky.

This game has some original ideas but they don't quite gel together and the overall impression is uninspiring.

GM.

## Fighting Warrior

McBauer House £5.95 cassette £14.95 disk C&A

3
 5
 6
 5



AS PRINCIPLES EVERYWHERE want to die, yours has managed to go and get himself kidnapped and you set off to

rescue her. The setting is ancient Egypt and in order to achieve your quest, you have to battle against assorted

creatures such as humans with jackal heads and winged demons.

Both you and your opponent try to hack each other to bits with swords. The amount of damage you can sustain is depicted by a number of arrows at the bottom of the screen and a successful strike reduces this total by one. When it reaches zero, your adversary dies or you lose one of your five lives.

Combat itself gives you a choice of three aggressive and four defensive manoeuvres. You can aim a high, medium or low blow and can jump, duck

and move forwards and backwards. In practice, the battles tend to degenerate into a slugging match with both sides trading wild and trading blows. After the combat, you get the chance to take a swipe at the magic vase that appears. This may increase or decrease your strength, take you to another zone or force a battle with a god.

Fighting Warrior lacks any lasting appeal. Every battle is much the same as the last and there is only a limited range of movements available.

There are better combat games on the market. GM.



## Journey

CRIE £5.99 CMI • Jystick

5 5 3 3



KEEP YOUR HANDS FREE AND hold on to the sides of the vertical shafts as you climb down, otherwise you could

come to a disastrous end.

The adventure must collect 11 treasures from the centre of the earth and bring them to the

surface. Dragons, magnetic fields, dynamite, detonators etc. hinder your search for the treasures. The player can escape from the samplings by climbing up or down a shaft. There are guns which can be used to kill deadly vampire bats and oxygen pills which can be taken when you enter a gas chamber. To pick something up you must stand over it and push up on the joystick. If you get too heavy you can drop something by pushing down on the joystick.

This is yet another arcade strategy game of the Dungeons and Dragons ilk. The special effects show you conversational to your death

and those used in the transporters are original and good. There are two speed levels, fast and normal. The documentation is adequate and full playing instructions can be found on the back of the cover. The player has three lives and starts each time at the surface. There are different levels of skill the first one being that of a business mail.

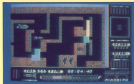
At first I was continually having to start over again after conversational to my death. I found the trick is to hold on to the sides as you climb down. I recommend this challenging game to anyone who wants a few hours of fun.

J.L.

## Thunderbirds

Firebird • Super Silver Range £1.95

8 8 9 5



A TEAM OF ARCHAEOLGISTS have got themselves trapped in an Egyptian tomb. They send out a mayday distress signal which is picked up by International Rescue. Thunderbirds are go.

You control both Thunder-

birds 1 and 2 (the fire button toggles control between them) and you have to find your way through a maze of passageways inside the tomb. As in all the best tombs though, it contains a huge number of traps. These come in two forms - huge stones that block the passageways and guardians such as mummies and spiders that try to stop you.

The main problem is the blocks. These are of three types. TB1 can only move blue blocks, TB2 green blocks and both can shift yellow blocks. The maze has been very ingeniously designed and it requires considerable planning to get through. Frequently, you

think you have solved a problem only to find that one of your walls is blocked in. To get to the correct position to shift certain blocks, one of the TBs is quite likely to have to detour through three or four other chambers first.

TB2 can carry certain equipment and a menu lets you choose what to take. These items may help you in a later stage of the game but you have to find out how and where. (see p. 16)

Thunderbirds, although not graphically brilliant, is an excellent game which is guaranteed to keep your grey cells ticking over. TAB Virgil.

C.R.H.

## Quest for the Holy Grail

Masterdisk £1.99 CMI

1 1 4 5



"MONTY PYTHON WITH chips" reads the tin. This supposedly wacky adventure game should be served with

large dollops of green slime, bearing a striking resemblance to Scott Adams' original adventure games of yesteryear, this

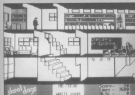
game tries to improve on them by using graphics, but not to any success.

So Lippin is on a quest to find the Holy Grail. The game starts off in a forge where the player meets a CND nut with a nuclear powered lamp and a key. He is able to go west, east, north, south, up or down. When Sir Lippin has been squashed by a falling tree or drowned by flying out or has betrayed some other taggled he restarts his quest in the forge again.

This value for money game offers the enthusiasm on a tight budget many hours of

exaggerating fun and a large vocabulary of four letter words. The writers have left all clues, instructions and commands of how to play to the players imagination or previous experience. Using 'map' yields "Yes, you'll need it". "No, I'm calling" or just plain "No". There are a few original results for some commands - try 'quit' for instance or use 'drop' and suffer the consequences once you are carrying the nuclear powered lamp! I found the language rule analysis to be and do not recommend it to anyone who wants a challenging and exciting game to play. J.L.

## School Daze Microsphere \$5.95 C&M



CONVERSIONS FROM SPECTRUM to C&M often disappoint me, but the plot of *School Daze* is so good that even the jolting scrolling action does not

deflect from the enjoyment.

It tells a tale with which everyone is familiar, the panic just before the school reports are issued. And it is a matter of

life and death to seal your report before the Headmaster sees it.

The hero is called Eric, a good name for a hero if ever I heard one, and he must obtain the secret combination of the schoolmaster's safe if he is to save his reputation. If the boy's name does not appeal to you, the names of the main characters can be changed at the start of play.

Like all schoolboys, Eric has lessons to attend and if he is caught wandering about during lesson time, or messing about generally, he will be given lines by passing teachers. If he is clever he will make sure that someone else is closer to the teacher when misbehaving because the blame generally falls on the nearest boy. If Eric answers more than 10,000 lines he is immediately expelled and the game starts again.

To find the combination, Eric must jump up and hit all of the shields which are hanging on the walls around the school. Some of the shields are too high for a mere schoolboy to reach so violent measures are called for in desperate circumstances. This may involve knocking down a fellow pupil or a schoolmaster and using them to give the necessary height to reach the shield.

When all of the shields have been set in motion, Eric must knock down each teacher in

turn to reveal a single letter of the combination. Unfortunately, the disliking old History teacher can only remember his if he sees his own date of birth written on the blackboard. Clues are given to the class during the game but if you guess wrong the nasty little rest of the class will split on you for writing on the board.

Once Eric knows the combination he must rearrange the letters into the correct order, he only knows that the headmaster's code first. The guess is written on a clean blackboard and then he must rush to the staffroom, jump up in front of the safe but if it doesn't open he must go and guess again.

Reviews of the original *Daze* ran and the game because the shields must be hit again to stop them flashing. Skillful use of a catapult can always help with hitting the shields but this does run the danger of getting lines.

The screen is a love of activity but it's a pity that more was not made of the C&M capabilities. A touch of blatant action creep in with all of the characters being made but in the tradition of Tom Brown's Schooldays, and Billy Bunter I shall overlook this and not put Microsphere in detention.

The old saw about these being the happiest days of your life is laid bare by this, the most traumatic game of your life.

ELB

## Willow Pattern Patched £15



MOST PEOPLE WILL BE MOANING about the design. The design stage have ruined a most off plans with the willow pattern

design on them. The design represents an ancient Chinese fairy tale and Firebird's latest

game is an arcade adventure based on that story.

You play a mere clerk who is in love with a princess. The trouble is, she is promised to a merchant so you decide to break into her palace and rescue her. To do that, you must find your way through a maze, find certain objects, overcome the palace guards and then escape to a boat.

In order to defeat a guard, you must throw a sword at him before he throws one at you. You can only carry one sword at a time and so you will have to backtrack a lot, find a sword can be found lying about or you can entice a guard to throw

one and then dodge out of the way.

Occasionally, you have to cross a bridge which you do by leaping from stepping stone to stone. It's not quite that easy as these giants try their hardest to knock you over and so timing your leaps becomes crucial.

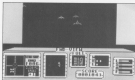
The graphics in *Willow Pattern* are exquisite with pictures of Chinese temples, bridges and trees. My first impression on playing it was that it was a Saker Wolf look-alike but the way that the game plays makes it totally different. At £15 it is excellent value for money.

G.R.H.

## Space Hunter

Mastertronic £1.99 C64 + joystick

 2
  6
  6
  5



THE RED ALERT FLASHES AND you prepare to defend yourself against the waves of fighters that threaten to destroy you.

The inhabitants of Earth are starving and none of the space fleet pilots have returned from their mission in search of food.

As a last desperate attempt to save the Earth you, a Rookie pilot, have been sent to capture the food transporters before the aliens.

The position of the target ship, which can be changed by the player, is indicated by a flashing dot on the radar. Use the warp drive or forward thrust to speed you to the target ship, as entering brings hordes of hostile fighters. Drive close enough, an approach speed of one or two will automatically teleport you into the ship. Now with the aid of a jet pack you can scan your search for extra 'warp drives', flashing food units, more fuel

and shields. Beware of the sneaking creatures which will deplete your energy and kill you if you collide with them. Once you have collected all you can, exit and find the next target ship.

For a cheap game styled on Elite this is not at all bad. To progress through the 11 levels from Rookie to Space Hunter will not take an experienced Elite-ist too long. The documentation, as in many Mastertronic games is not very comprehensive. The graphics are realistic and the music is pleasantly relaxing after your ordeal with the fighters.

ILL

## Chimera

Firebird - Super Silver Range £3.95

 4
  7
  7
  7



THAT THE CRAFT ORBITING the earth is hostile is undeniable. It also appears that someone is going to have to

find out how much of a threat it represents and if possible destroy it. If you escape with your life, so much the better!

As the airlock shuts behind you, you look around and see that you are in a large chamber with exits left and right and an exit in front of you blocked by what appears to be a giant microchip. The rooms are depicted in 3D block graphics, reminiscent of Ultimate's Alien 8 and Knightmare for the Spectrum. The effect works well.

Detraction of the ship requires a four stage sequence and your only clue is that the first stage requires a scanner so it seems reasonable to get off looking for one. There is nothing much to stop you as you explore - nothing to shoot

- but certain areas are restricted to you at the start and kill you if you try to enter them. Time however is against you and your supplies of food and water soon disappear and have to be replenished as you find them.

You die if either level reaches zero. Objects are frequently hidden behind corners 'that you can't see' and so detailed exploration is essential. A scrolling message keeps you informed of what's going on.

Chimera is an enjoyable arcade adventure but it lacks the sophistication of some of its bigger brothers.

C.B.H.

## Friday the 13th

Danmark £3.95 C64 + joystick

 7
  8
  8
  7



FRIDAY THE 13TH IS BASED on the eighth-6 movie of the same name. It is a tale of twisted revenge as homicidal maniac

Jason tries to avenge his mother's murder.

His wrath is turned against the innocent holiday makers

on the shores of Crystal Lake and the field of action comes to screens with blood. In addition there are three buildings: a church and a barn (four screens each plus a bungalow (3) screens). This gives Jason plenty of rooms to cleave.

The computer selects a character for you to play and it is your duty to look after the other players by handing them all into a room where you have placed a sanctuary cross.

At times point you may have to arm yourself with one of the weapons lying around and tackle Jason in combat. If you succeed in killing him don't feel too secure, just when you

think you have seen the last of him, up he pops again looking pretty cut-up about his last encounter.

The game is quite challenging to play and the graphics are blocky but pleasant. I don't really think that the quality of the game will really have as much sway with sales as the horrific subject matter. The playing instructions give hints on how to get the most from the horrifying screaming sound effects.

A gory story for the blood-thirsty buyer and just to quickly check that you get two booming blood capsules with the game.

I.D.

## Enigmasforce

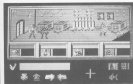
Byond Software DS/32 C64 + joystick



THE 8 TEAMS ARE BACK WHEN Enigmasforce try to overcome the deadly General Zoff. This is the expected sequel to Madworld and the gameplay is even more advanced than before.

Having captured Zoff in the previous game, four members

of the force were accompanying Zoff to his trial when their ship crashed on team member Syllik's home planet after collision with a gravity mine. The game starts in the Enigmasforce members regain consciousness and discover that Zoff has made off.



Syllik's insectoid people have enough to worry about with an attack from reptiloids so the team has to make the help of the insectoid commander. This involves the team in much frantic searching and the game allows you to disperse your force as you see fit. Each member can be located and relocated whenever necessary and the activity selection panel on the lower half of the split-screen is used to issue commands.

Enigmasforce consists of Zark Monitor the team leader, Syllik the strong warrior, Sevisia, brain the mathematician and locksmith, and last but not least Maul the combat Droid.

As the team goes from location to location they can pick up objects which are lying there but care has to be taken to ensure that each player's special skills are catered for. When the commander has been located, the reptiloid zone must be searched for the only visible spacecraft on the planet. Zoff is also searching and must be stopped and captured before Republican destruction bugs arrive to blast the planet to smithereens.

The activity panel is fairly complex and it takes a little time to familiarise yourself with it. At the top is a picture of each team member and if the crosswire selector is placed over one of these panels and the fire button pressed then you are

immediately shown that team member's current location. You can now use the arrow keys to dictate a direction for that character to take or select a sequence of actions for them to perform. At first it is relatively simple to use the panel but as the game progresses and the action heats up, you have to keep your wits about you and your joystick on the move.

Action commands are like an icon driven version of an adventure command. For example, to get Maul to pick up explosives you first select Maul's icon panel, then check the inventory of objects around to see that the explosive icon is there and place the crosswire over the pickup icon. If the button is pressed when the crosswire has been moved over the explosive icon, it will move across into the panel showing the list of objects carried by Maul.

As you can guess the game and the graphics are very sophisticated and just many similar multi-screen games to share, it just shows what can be done with the 32 in the hands of an intelligent programmer. Smooth animation of each character's movement adds cartoon realism to the game and the excitement of the many skirmishes should hold your attention for hours.

IDS

## The Human Race

Masterbyte EL/39



THE HUMAN RACE TOOK 500,000,000 to develop plans and believe me it will take you as many hours to complete this

game. The idea is quite simple: you have to develop a rather ugly and stupid ape into a man, remember Darwin's theory!

This is achieved by completing each screen after which the ape slowly develops its manly features.

The game begins in prehistoric times complete with a large dinosaurs, dragonfly, pterodactyl and lots of bananas. Your task is to get the ape to the largest banana while avoiding contact with the roaring dinosaurs. The route to the banana appears very straightforward but it requires some thought if you are to reach it. Once completed chapter two begins.

Here the ape is sent forward to the age of lava and fireballs. His task is to reach the end of a

moving pathway without falling off or getting hit by a lurking fireball. This screen belongs to something like Journey To The Centre-Of The Earth!

As the game continues the tasks become very difficult but still enjoyable and certainly provide a good challenge.

The graphics aren't the best I've seen and there is a rather annoying delay every time a life is lost but generally the game provides excellent value for money.

Go ahead and buy it and watch out for a lovely koinkorsh that would be worthy of any Tartan!

IDS

**You don't have to be mad to work here, but it helps. Marie Curry visited Ian Stewart at Gremlin Graphics.**

THERE'S A LITTLE GREEN MAN ALIVE and well and living in Sheffield. This pre-coloured gremlin has made his home there since June 1984 and the climate seems to agree with him.

Gremlin Graphics is a small compact outfit run by the ebullient Ian Stewart who defines himself as the inspiration behind many of his company's products. Entering the micro industry through last year's first computer outlet in Sheffield, he soon realised that there were opportunities for a bright young man in this youthful business.

Inspired with this confidence, Ian opened Sheffield's first specialist computer shop aptly named, Just Micros. The venture took off and Just Micro did just fine. It was less than a year later that the gremlin got into the works.

In the summer of 1984, Ian and partner, Kevin Not Barns, took the plunge, caught the gremlin and took him on some headed notepaper to form Gremlin Graphics. Since this was done, there was no going back and wanted. Monty Mole, Gremlin's first game, was launched amid a blaze of publicity. The misadventure was then entering the bag and not so hot summer and the Gremlin Graphics boys see the potential of Arthur Scargill's activities as material for a computer game storyline. Because of its topical content the game was treated inside coverage on both national television and radio and became a chart success giving the way for two subsequent Monty games.

Ian Stewart believes that a major reason for Gremlin Graphics's continued success in the production of popular games is the connection which is closely maintained with just Micros. All Gremlin's games and extensions travel on unsuspecting members of the public who innocently venture into the shop. "Customers are excellent bug fighters," commented Ian. "Our programmers may test a game for days without finding anything, then one of the kids will come across a bug in a few minutes of play."

The Gremlin outfit listens very carefully to the comments of the customers who try out new products. Ian briefly summed up the Gremlin policy in this area: "If we don't get the reaction we want then we scrap it." Simple but effective.

Many Gremlin game ideas are built up around a central central character and this seems to have become a successful technique. Names such as Potty Pigeon, Sam Snot Safebreaker and Thing on a



# Gremlin GRILLING



Spring came to mind. "We like to have a game around a particular character to stimulate the imagination of the player," said Ian. "We tend to concentrate on arcade games because they give an outlet to the sense of humour which is a ruling principle at Gremlin. Whenever I am, I'm always thinking of new ideas for games."

There are only four full time programmers at Gremlin, what work is done by about 10 irregular freelancers. Pete Harrap who wrote the original Monty Mole is now a permanent member of staff. His association with Ian Stewart began when his Spectrum broke down and just Micros lent him another until it was repaired, Ian said. "Pete was a customer in the shop and we realised he had great programming ability. We lent him the Spectrum and it just went on from there." However, Sheffield natives with debutant computers shouldn't rush round to try and get a loan of a machine.

The full time programmers at Gremlin are now collaborating to produce a series of games based on the popular role-playing books. The Way of the Tiger, the books revolve around the adventures of Avengers, a Ninja warrior. Work on the series is progressing fast and the first game should be available in February. Although programmers at Gremlin have always worked together to a certain extent this is the first project into which the team have plunged their joint efforts from its inception and the co-operation between them has been extremely successful. Praising his boys, Ian said: "Kudos which need improvement can be sorted out through collaboration. There's never a cross word between them." There will be four games in the series when it is

completed and they will all follow a similar story line to the books.

Another new game, Bouncer, features the incredible antics of a bouncing tennis ball. It's certainly a different concept for a game as the ball bounces round you out of the screen and then bounces back into the distance. Gremlin's programmers have even got pin on it! According to Ian the game comes in a value for money pack with another game, entitled Metabolix, on the back of the cassette.

C-16 and Plus/4 games are an important part of the Gremlin range and owners will be pleased to know that Gremlin has no intention of discontinuing this commitment to these machines. There are probably around 80,000 C-16 and Plus/4 owners in the country now and there are very few software houses which take notice of them. Looking at the situation in the cold light of day Ian remarked: "The less people that produce games for the C-16 and Plus/4, the better for Gremlin." A recently released compilation contains four games for 10.00 so the thrice neglected users can really get rapping.

Ian obviously believes he's got his policy right but he stresses the fact that above all the computer industry is a lot of fun. "The main strength of Gremlin is the atmosphere in the firm. This makes for a good working situation and consequently good products," he remarked almost dreamily and by the tapping notes from next door and the blast of electronic music from a neighbouring office.

When asked to sum up Gremlin as a software house, Ian makes the place sound more like a lunatic asylum. "Basically we're a barmy lot up here."



# Missives

Every program that we publish in this magazine is very thoroughly tested before we print it.

The listings in the magazine are not typewritten in any way. What we actually do is get a printout of the program which is then placed on the page as artwork. This leaves very little chance for errors to occur.

Obviously errors do creep in sometimes. When they do we always publish corrections in the magazine. Corrections don't usually appear in the next issue of *Your Commodore* but sometime after that. This is because when one issue is on sale another is just about ready to be printed.

Most of the time the errors are made by the person typing the listing into the machine. We make many times go back to check them for errors closely enough. In order to help you with this we will often to send a time computer listing to anyone having problems with a program if they send us a stamped self-addressed envelope and state clearly what they require.

In the near future we are hoping to start a software service where all of the programs in a particular issue will be available on cassette but we have no firm date for this at yet.

In the meantime, a couple of features that will try to give you some hints about debugging programs have been commissioned. And, being honest, *Audit* finding a program is an extremely good way of learning about programming. Perhaps we should start a spot the deliberate mistake page!

Coming back to the System 64 program, it appears that the programmer made a few mistakes when he wrote the program. One only deletes the name and not the address. This is done in the *Sort* routine. This is the type of mistake that it is not always possible to spot so please make sure that you give any programs sent to the magazine a thorough testing before you stick them in the post. Anyway here are the lines that will need to be added to delete a whole record.

```
1562 LET L261=L261+1 LET L261=1
1563 LET L261=L261+1 LET L261=1
1564 LET L261=L261+1 LET L261=1
1565 LET L261=L261+1 LET L261=1
1566 LET L261=L261+1 LET L261=1
```

## Too Many Ads?

I am writing to complain about the recent change which has occurred since the merging of *Your Commodore* and *Your 64*. It has come to my attention that there has been a substantial increase in the amount of advertisements, at least 25% in the December issue. As I subscribe to your magazine I am worried that the amount of advertisements may increase even further in the future.

I would like to see in the magazine an extra page of *Flippo* (in *Amiga*) in exchange for one less page of *Sense of Adventure*, as I feel that *Flippo* is far more interesting.

A suggestion I have for your monthly competition is, perhaps, that you should ask people to send in the best one line program or the best game or utility using no more than 100 lines since a spot the difference competition does not involve the use of a computer in any way.

I hope that the points I have brought to your attention will help to make your magazine even more absorbing.

**Raymond Webb, Garsling**

Thanks for your comments, Raymond. It's always interesting to find out what people think of the magazine. Your main worry seems to be concerning advertising. This varies greatly throughout the year and tends to rise at Christmas because manufacturers want to make the most of the extra money we all spend. Advertising takes a large part of our income and it is therefore essential that we carry a certain amount in order to keep up the standard of the magazine as a whole.

As an arcade game fan, we can bring you some good news. In the near future we will be taking some more ideas from *Time* to improve our arcade coverage. However, adventure players needn't worry because you won't suffer from this at all.

As for your comment on our competitions, Raymond. We do try and run the occasional different one, but the beauty of a spot the difference competition is that anyone can enter, and they needn't have any knowledge of programming to do so. Since the points are usually games of some sort, then it seems illogical to set a problem which only programmers can answer.

## Yet another bundle of replies to your letters.

### Problems, Problems

I WAS DELIGHTED WHEN THE PROGRAM, System 64 by J. A. Wolfe appeared in the August issue of the magazine. I was able to enter all the addresses of my associates and it worked well. That is to say until this week!

I wanted to amend the address of a colleague who had moved house. So, using option 1 ('Delete Record'), I cancelled the old address and reentered the new address on the end of the list of addresses.

Imagine my disappointment when I printed out a fresh list of addresses to find that every single address after the one I had deleted now had the wrong name. It appears that the deletion in your program only erases the name and not the whole address. I have studied the programming but have been unable to amend it. Can you help?

**B M Peasmore, Sidmouth**

I have typed in the Home Accounts program from your magazine. However I cannot seem to get it to work.

I have checked my program against the one printed in the magazine and can't find any errors. I haven't seen any corrections printed for this program but I believe that the error is yours and not mine.

Why don't you check your programs before you print them in the magazine as it would no doubt solve many problems! **A. Sherwood, Brillingham**

Every page delivered to the *Your Commodore* office is guaranteed to have quite a large number of letters similar to the ones above. So it is probably worth making a few points clear.

# LANGUAGE

## TOOL

## LAB—

David Janda takes control and guides you through Commodore's Pilot package.

THIS MONTH'S LANGUAGE LAB is rather different from those in previous issues. Presented here is a brief introduction to a language that is very popular in the educational field in America. No, it's not Logo, but Pilot, and in my opinion, it would be just as popular if it was given the amount of attention it deserves. Commodore Pilot is the only package which is being reviewed, and the surprising thing is that the Commodore version of standard Pilot (known as *commodore Pilot*) includes many facilities not found in the standard.

It is worth pointing out that although the language is designed to be used by educators to write interactive educational programs, it does have its attractions for the programmer. The reason I say this is because of pattern matching.

It can be said that the job of a programmer is to solve the problems at hand. In educational programs this can be quite difficult as there is a lot of user input which has to be interrogated. This is not so

easy, and programmers normally find themselves spending a lot of time writing routines that interrogate user input. This can distract them from the job at hand — i.e. solving the problem!

Pilot incorporates a number of tools that facilitate the interrogation of user input, thus leaving the programmer to get on with the main task. These tools can be applied to problems associated with string manipulation and pattern matching.

### Pilot — The Language

Pilot is a computer assisted instructional (CAI) language, which is designed for teachers to produce educational programs — *conversate* as the manual terms it. The idea behind Pilot is that it should be easy for teachers to produce programs that interact with the student on a question and answer level.

The version which is supplied on disk will run Commodore Pilot programs, and do a lot more since extensions have been added. These allow for the use of colour, graphics and sprites. An example is that text windows may be set up to allow for questions and answers to be displayed on different sections of the screen.

Graphics are entered for with commands to allow lines to be drawn/printed as well as filled in with colour. A group of sprite and colour commands is also included, and the extensions facilitate the use of the advanced features of the C64.

The syntax of Pilot is very simple. Pilot instructions consist of several parts. First, there is a one-letter opcode (at which there are 30). The opcode is then optionally followed by a modifier which changes the way the opcode is going to be executed. Conditions can then follow and they can determine whether the instruction is to be carried out. This is best described with an example:

T1A=50; Correct, the answer is 5.

Here, the opcode T means print something, but the screen is first cleared with the modifier S. The condition is that the answer A equals five, and if so the text in the field (everything following the colon) will be printed.

Unlike Basic, Pilot does not require strings to be enclosed in quotes. Instead, the string variable or literal is placed after the separator. This very simple Pilot program demonstrates this:

T: This will be printed

This will simply display "This will be printed" at the current cursor location. However, a return is also printed after each occurrence of the T command. It is possible to 'hold', or keep the cursor on the same line by using the H modifier.

list: What is your name

This will display the message, and any answer will be entered on the same line (it is important to leave two spaces after the message).

### Pilot Data Types

Maths in Pilot is integer only which is a bit of a restriction. The range is between -32768 and +32767. Another restriction is that only 26 variables are allowed for.

Performing calculations is done with the computer instruction which takes the form of C. Variables assignment is performed after the separator and printing the values of variables requires a P character to precede the variable name.

C:A=2 — assigns A with 4

C:B(2-3)\*5(4) — assigns B with 18

3 Here is a sum...

T1A: 2+2

C:A=2+2

P1A

3 There, the answer is 1A

strings are handled in a very flexible way. But, as mentioned Pilot only has 26 variables available so it is not possible to have a string and a numeric variable of the same letter.

Before a variable can be used as a string, it must first be dimensioned with its maximum length. Pilot allows a maximum string length of 255, and the command used to dimension string is D. The \$ character is used to identify that the variable is a string, but this is not absolutely necessary.

D:A\$(10)

The computer command is used to assign a string variable with a value. In this case, it is necessary to use quotes. When using the T command to print the contents of a string variable a \$ character is used to precede the variable.

D:A\$(14)

C:At "Your Commodore"

T:This magazine is called \$A\$

The C command is very flexible when it comes to string assignment. Strings can be assigned with sub-strings, concatenated and so on. Indexing is allowed which can be used on the source or object string.

D:A\$(10)

D:B\$(20)

C:G "This is funny"

C:G\$ "He is not very funny"

C:G\$(3+G\$(11,4)

T:\$A\$

Would print: "This is very funny".

Concatenation is also possible using the + operator.

D:A\$(4)

D:B\$(7)

D:C\$(10)

C:A\$ "Hello"

C:B\$ "realizers"

C:C\$=A\$B\$

T:HC

Would print: "Hello realizers".

Getting user input into the micro is very simple in Pilot. A pre-defined input buffer called \$I\$ is used to store user input. It works like this:

T:Hello, who are you?

A:

T: Please to meet you \$I\$

Notice that like ordinary string variables, the input buffer requires the \$ to be prefixed to the buffer name when it is being printed.

The & is the accept command, and it can work with numeric and string

variables as well as the input buffer variable:

D:A\$(10)

T:\$I\$ What is your name?

A:\$A\$

T: And how old are you \$A\$ ?

A:\$I\$

T: You are 18 years old then \$A\$

The problem with user input is that you don't always get it in the format you wanted. Some people would enter their names in upper case, some lower, and some as a mixture of both. This can be a real headache especially if the input is to be processed. Pilot provides a problem command PR that allows input to be 'converted' into a specified choice. PR-U will convert all input to upper case, PR-L to lower and PR-S will strip any input of spaces. PR-X will reset the options.

## The Clever Stuff

Pattern matching is achieved with the match command M. Assuming we wanted to check that the user reads this mag, we could pose the question and process the answer using this program:

T:What Commodore magazine do you read?

M:Your Commodore

A:

Now, if the answer entered was "I read a magazine called Your Commodore which I think is great", believe it or not a match would be made. This is because Pilot does the hard work involved in searching (called window searching) though the user input checking to see if there is a match.

Problems with this are that the user may enter the answer in upper or lower case. To still get a match the PR command would be used before the match command to convert the input.

Match used with the \$ modifier will even accept answers that are spelt incorrectly! The S modifier will accept an answer even if one letter is wrong, or if a pair of characters have been reversed - very handy!

More flexibility is allowed with the \* and & characters when used in the match command:

M:Comer\*dore

This simply means 'accept any letter in place of the \*'. The & means any number of characters.

## Summary

There are many, many other features to the Pilot language. Jumping, subroutines,

multiple choice tests, even hinting is possible. All these features make Pilot a very practical tool for educational programming.

The language does suffer in some areas though. The restriction on the number and length of variables is a serious one, as is the lack of floating point maths, but these faults are common to the Pilot standard, and are not unique to Commodore's implementation.

Even though the language is very powerful, it is not very hard to learn. Commodore Pilot has 26 commands with modifiers and because the syntax is very simple, it is quite possible to write complex programs in a very short time.

I would strongly recommend Pilot to anyone who wishes to write programs that involve processing interactive answers. The features available in the language enable the programmer to 'get on with the job'.

## Commodore Pilot

Commodore Pilot is supplied on disk only, together with a very good 111 page tutorial/instruction manual. Two versions of the Pilot interpreter are supplied on the distribution disk: a development version which is used to write, edit and run Pilot programs, and a run only version that is identical except programs can only be loaded and run.

Other files on the disk include three demonstration programs, a simple sprite editor written in Pilot, and a Pilot program that enables the user to experiment with sounds on the C64.

The actual Pilot package has four modes of operation. First there is the edit mode which is used for program creation and editing. The run mode is for running the program, and the command mode is used for loading, saving and printing programs. Finally, the immediate mode (which is similar to Basic's immediate mode) allows the programmer to experiment with Pilot by trying out Pilot commands one at a time. This mode is very handy when learning the language.

Graphics on a 128x256 grid are catered for. Pictures can be plotted and rotated, lines drawn and the graphics origin changed. Both text and graphics can be freely mixed and a split-screen command allows the screen to be divided between graphics/text output and prompts/user input.

Other features include sound, sprites and user definable characters, although I must say that these could have been implemented in a more friendly way.

Commodore Pilot not only conforms to the standard common Pilot, but also includes many new features (some of which I have mentioned). The package is an ace one to use which is a blessing. Highly recommended!

# PROGRAMMER OF THE YEAR

Continued

## This month's entry is

**Spike, an excellent**

**game by Shane**

**Stevens.**

A LARGE NUMBER OF GAMES have been entered for the Programmer of the Year Competition. Spike is certainly an above average entry. It is definitely worth the effort of trying it in.

In the game you find yourself as Spike travelling around a Power Grid. Hidden somewhere within the grid is your trusty CIA. Your job is to find it.

Of course, life isn't easy and the Sparks brothers are out to get you. The number of sparks charging around the grid depends on the level at which you choose to play; there are nine in all.

Full playing instructions are included in the game so there is no point giving them here.

## Getting It All In

Spike is in two parts, the first part is in Basic and should be typed in and saved on to tape or disk. Make sure you read the page that tells you all about our method of printing listings before you start (it don't want [SHAME] - Ed).

If you are using a cassette then make sure that you change the

```
LOAD "SPIKE",1,1
```

in line 40 to:

```
LOAD "SPIKE",1,1
```

Once you have SAVED the Basic you can then tackle the machine code. Yes, I know that there's a lot of it but we have tried to make it as neat as possible for you.

Elsewhere in this magazine you will find the Your Commodore Easy Entry program. You should type this in and save on something safe. You will need this for most of the machine code programs in your Commodore. RUN this and follow the instructions with the Easy Entry article. Don't forget each line is checked as you type it in and you can SAVE what you have entered at any time.

Make sure you SAVE it before you attempt to RUN it.

Spike should be SAVED straight after the SPIKE LOADER and is SAVED between the following locations:

Start Address : 10000  
End Address : 17000

Remember to press F1 in the Easy Entry program to activate the SAVE routine, and

make sure you save the program with the name SPIKE.

## And On We Go

Now that you have both parts saved on tape you simply have to LOAD and RUN the SPIKE LOADER program, this will automatically LOAD the second part and the game will start to RUN.  
Have fun!

### Program: Spike Load

```
1 REM SPIKE BY GAME STEVENS
2
3 IF NOT THEN GOTO 50
4 A=+1:GOTO "SPIKE",1,1
5 REM CHANGE TO :1,1 FOR
6 CASSETTE
7
8 G=0:GOTO:FOR I=0 TO 5:G=
9 READ G:POKE I,G:NEXT
10 DATA 76,22,1,1,1,1,76,164,
11 175,18,1,1,1,75,186,48,1,
12 1,1,1,75,186,1,15,211,10
13
14 FOR B=40:160 TO 17000
15 READ B:POKE B,B:NEXT
16
17 DATA 170,224,1,174,
18 200,1,170,224,1,141,1,212,
19 140,18,212,86,
20
21
22 DATA 226,224,1,204,225,1,
23 204,225,1,12,1,175,226,
24 224,1,226,225,1,226,224,1,
25
26
27
28 POKE 192,17:POKE 193,48
29 POKE 194,10
30
31 POKE 5:10,10:FOR T=0 TO 1
32 NEXT:POKE 5:10,10
33 POKE 5:10,10
34
35 FOR T=1 TO 10:NEXT
36
37 POKE 5:10,10:POKE 5:22,1
38 POKE 5:10,10
39 POKE 5:10,10:FOR T=0 TO
40 NEXT:POKE 5:22,7:NEXT
41
42 POKE 5:10,10:POKE 5:22,70
43 POKE 5:10,71:POKE 5:22,2
```

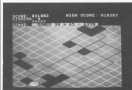
```
POKE 5:10,10:FOR T=1 TO
44 NEXT
45
46 POKE 5:10,10:POKE 5:10,
47 POKE 5:10,10
48
49 POKE 5:10,10
50
51 POKE 5:10,10:POKE 5:10,10
52
53 POKE 5:10,10:POKE 5:10,10
54
55 POKE 5:10,10:POKE 5:10,10
56
57 POKE 5:10,10:POKE 5:10,10
58
59 POKE 5:10,10:POKE 5:10,10
60
61 POKE 5:10,10:POKE 5:10,10
62
63 POKE 5:10,10:POKE 5:10,10
64
65 POKE 5:10,10:POKE 5:10,10
66
67 POKE 5:10,10:POKE 5:10,10
68
69 POKE 5:10,10:POKE 5:10,10
70
71 POKE 5:10,10:POKE 5:10,10
72
73 POKE 5:10,10:POKE 5:10,10
74
75 POKE 5:10,10:POKE 5:10,10
76
77 POKE 5:10,10:POKE 5:10,10
78
79 POKE 5:10,10:POKE 5:10,10
80
81 POKE 5:10,10:POKE 5:10,10
82
83 POKE 5:10,10:POKE 5:10,10
84
85 POKE 5:10,10:POKE 5:10,10
86
87 POKE 5:10,10:POKE 5:10,10
88
89 POKE 5:10,10:POKE 5:10,10
90
91 POKE 5:10,10:POKE 5:10,10
92
93 POKE 5:10,10:POKE 5:10,10
94
95 POKE 5:10,10:POKE 5:10,10
96
97 POKE 5:10,10:POKE 5:10,10
98
99 POKE 5:10,10:POKE 5:10,10
100
101 POKE 5:10,10:POKE 5:10,10
102
103 POKE 5:10,10:POKE 5:10,10
104
105 POKE 5:10,10:POKE 5:10,10
106
107 POKE 5:10,10:POKE 5:10,10
108
109 POKE 5:10,10:POKE 5:10,10
110
111 POKE 5:10,10:POKE 5:10,10
112
113 POKE 5:10,10:POKE 5:10,10
114
115 POKE 5:10,10:POKE 5:10,10
116
117 POKE 5:10,10:POKE 5:10,10
118
119 POKE 5:10,10:POKE 5:10,10
120
121 POKE 5:10,10:POKE 5:10,10
122
123 POKE 5:10,10:POKE 5:10,10
124
125 POKE 5:10,10:POKE 5:10,10
126
127 POKE 5:10,10:POKE 5:10,10
128
129 POKE 5:10,10:POKE 5:10,10
130
131 POKE 5:10,10:POKE 5:10,10
132
133 POKE 5:10,10:POKE 5:10,10
134
135 POKE 5:10,10:POKE 5:10,10
136
137 POKE 5:10,10:POKE 5:10,10
138
139 POKE 5:10,10:POKE 5:10,10
140
141 POKE 5:10,10:POKE 5:10,10
142
143 POKE 5:10,10:POKE 5:10,10
144
145 POKE 5:10,10:POKE 5:10,10
146
147 POKE 5:10,10:POKE 5:10,10
148
149 POKE 5:10,10:POKE 5:10,10
150
151 POKE 5:10,10:POKE 5:10,10
152
153 POKE 5:10,10:POKE 5:10,10
154
155 POKE 5:10,10:POKE 5:10,10
156
157 POKE 5:10,10:POKE 5:10,10
158
159 POKE 5:10,10:POKE 5:10,10
160
161 POKE 5:10,10:POKE 5:10,10
162
163 POKE 5:10,10:POKE 5:10,10
164
165 POKE 5:10,10:POKE 5:10,10
166
167 POKE 5:10,10:POKE 5:10,10
168
169 POKE 5:10,10:POKE 5:10,10
170
171 POKE 5:10,10:POKE 5:10,10
172
173 POKE 5:10,10:POKE 5:10,10
174
175 POKE 5:10,10:POKE 5:10,10
176
177 POKE 5:10,10:POKE 5:10,10
178
179 POKE 5:10,10:POKE 5:10,10
180
181 POKE 5:10,10:POKE 5:10,10
182
183 POKE 5:10,10:POKE 5:10,10
184
185 POKE 5:10,10:POKE 5:10,10
186
187 POKE 5:10,10:POKE 5:10,10
188
189 POKE 5:10,10:POKE 5:10,10
190
191 POKE 5:10,10:POKE 5:10,10
192
193 POKE 5:10,10:POKE 5:10,10
194
195 POKE 5:10,10:POKE 5:10,10
196
197 POKE 5:10,10:POKE 5:10,10
198
199 POKE 5:10,10:POKE 5:10,10
200
```

```
POKE 5:10,10:POKE 5:10,10
201
202 POKE 5:10,10:POKE 5:10,10
203
204 POKE 5:10,10:POKE 5:10,10
205
206 POKE 5:10,10:POKE 5:10,10
207
208 POKE 5:10,10:POKE 5:10,10
209
210 POKE 5:10,10:POKE 5:10,10
211
212 POKE 5:10,10:POKE 5:10,10
213
214 POKE 5:10,10:POKE 5:10,10
215
216 POKE 5:10,10:POKE 5:10,10
217
218 POKE 5:10,10:POKE 5:10,10
219
220 POKE 5:10,10:POKE 5:10,10
221
222 POKE 5:10,10:POKE 5:10,10
223
224 POKE 5:10,10:POKE 5:10,10
225
226 POKE 5:10,10:POKE 5:10,10
227
228 POKE 5:10,10:POKE 5:10,10
229
230 POKE 5:10,10:POKE 5:10,10
231
232 POKE 5:10,10:POKE 5:10,10
233
234 POKE 5:10,10:POKE 5:10,10
235
236 POKE 5:10,10:POKE 5:10,10
237
238 POKE 5:10,10:POKE 5:10,10
239
240 POKE 5:10,10:POKE 5:10,10
241
242 POKE 5:10,10:POKE 5:10,10
243
244 POKE 5:10,10:POKE 5:10,10
245
246 POKE 5:10,10:POKE 5:10,10
247
248 POKE 5:10,10:POKE 5:10,10
249
250 POKE 5:10,10:POKE 5:10,10
251
252 POKE 5:10,10:POKE 5:10,10
253
254 POKE 5:10,10:POKE 5:10,10
255
256 POKE 5:10,10:POKE 5:10,10
257
258 POKE 5:10,10:POKE 5:10,10
259
260 POKE 5:10,10:POKE 5:10,10
261
262 POKE 5:10,10:POKE 5:10,10
263
264 POKE 5:10,10:POKE 5:10,10
265
266 POKE 5:10,10:POKE 5:10,10
267
268 POKE 5:10,10:POKE 5:10,10
269
270 POKE 5:10,10:POKE 5:10,10
271
272 POKE 5:10,10:POKE 5:10,10
273
274 POKE 5:10,10:POKE 5:10,10
275
276 POKE 5:10,10:POKE 5:10,10
277
278 POKE 5:10,10:POKE 5:10,10
279
280 POKE 5:10,10:POKE 5:10,10
281
282 POKE 5:10,10:POKE 5:10,10
283
284 POKE 5:10,10:POKE 5:10,10
285
286 POKE 5:10,10:POKE 5:10,10
287
288 POKE 5:10,10:POKE 5:10,10
289
290 POKE 5:10,10:POKE 5:10,10
291
292 POKE 5:10,10:POKE 5:10,10
293
294 POKE 5:10,10:POKE 5:10,10
295
296 POKE 5:10,10:POKE 5:10,10
297
298 POKE 5:10,10:POKE 5:10,10
299
300 POKE 5:10,10:POKE 5:10,10
301
302 POKE 5:10,10:POKE 5:10,10
303
304 POKE 5:10,10:POKE 5:10,10
305
306 POKE 5:10,10:POKE 5:10,10
307
308 POKE 5:10,10:POKE 5:10,10
309
310 POKE 5:10,10:POKE 5:10,10
311
312 POKE 5:10,10:POKE 5:10,10
313
314 POKE 5:10,10:POKE 5:10,10
315
316 POKE 5:10,10:POKE 5:10,10
317
318 POKE 5:10,10:POKE 5:10,10
319
320 POKE 5:10,10:POKE 5:10,10
321
322 POKE 5:10,10:POKE 5:10,10
323
324 POKE 5:10,10:POKE 5:10,10
325
326 POKE 5:10,10:POKE 5:10,10
327
328 POKE 5:10,10:POKE 5:10,10
329
330 POKE 5:10,10:POKE 5:10,10
331
332 POKE 5:10,10:POKE 5:10,10
333
334 POKE 5:10,10:POKE 5:10,10
335
336 POKE 5:10,10:POKE 5:10,10
337
338 POKE 5:10,10:POKE 5:10,10
339
340 POKE 5:10,10:POKE 5:10,10
341
342 POKE 5:10,10:POKE 5:10,10
343
344 POKE 5:10,10:POKE 5:10,10
345
346 POKE 5:10,10:POKE 5:10,10
347
348 POKE 5:10,10:POKE 5:10,10
349
350 POKE 5:10,10:POKE 5:10,10
351
352 POKE 5:10,10:POKE 5:10,10
353
354 POKE 5:10,10:POKE 5:10,10
355
356 POKE 5:10,10:POKE 5:10,10
357
358 POKE 5:10,10:POKE 5:10,10
359
360 POKE 5:10,10:POKE 5:10,10
361
362 POKE 5:10,10:POKE 5:10,10
363
364 POKE 5:10,10:POKE 5:10,10
365
366 POKE 5:10,10:POKE 5:10,10
367
368 POKE 5:10,10:POKE 5:10,10
369
370 POKE 5:10,10:POKE 5:10,10
371
372 POKE 5:10,10:POKE 5:10,10
373
374 POKE 5:10,10:POKE 5:10,10
375
376 POKE 5:10,10:POKE 5:10,10
377
378 POKE 5:10,10:POKE 5:10,10
379
380 POKE 5:10,10:POKE 5:10,10
381
382 POKE 5:10,10:POKE 5:10,10
383
384 POKE 5:10,10:POKE 5:10,10
385
386 POKE 5:10,10:POKE 5:10,10
387
388 POKE 5:10,10:POKE 5:10,10
389
390 POKE 5:10,10:POKE 5:10,10
391
392 POKE 5:10,10:POKE 5:10,10
393
394 POKE 5:10,10:POKE 5:10,10
395
396 POKE 5:10,10:POKE 5:10,10
397
398 POKE 5:10,10:POKE 5:10,10
399
400 POKE 5:10,10:POKE 5:10,10
401
402 POKE 5:10,10:POKE 5:10,10
403
404 POKE 5:10,10:POKE 5:10,10
405
406 POKE 5:10,10:POKE 5:10,10
407
408 POKE 5:10,10:POKE 5:10,10
409
410 POKE 5:10,10:POKE 5:10,10
411
412 POKE 5:10,10:POKE 5:10,10
413
414 POKE 5:10,10:POKE 5:10,10
415
416 POKE 5:10,10:POKE 5:10,10
417
418 POKE 5:10,10:POKE 5:10,10
419
420 POKE 5:10,10:POKE 5:10,10
421
422 POKE 5:10,10:POKE 5:10,10
423
424 POKE 5:10,10:POKE 5:10,10
425
426 POKE 5:10,10:POKE 5:10,10
427
428 POKE 5:10,10:POKE 5:10,10
429
430 POKE 5:10,10:POKE 5:10,10
431
432 POKE 5:10,10:POKE 5:10,10
433
434 POKE 5:10,10:POKE 5:10,10
435
436 POKE 5:10,10:POKE 5:10,10
437
438 POKE 5:10,10:POKE 5:10,10
439
440 POKE 5:10,10:POKE 5:10,10
441
442 POKE 5:10,10:POKE 5:10,10
443
444 POKE 5:10,10:POKE 5:10,10
445
446 POKE 5:10,10:POKE 5:10,10
447
448 POKE 5:10,10:POKE 5:10,10
449
450 POKE 5:10,10:POKE 5:10,10
451
452 POKE 5:10,10:POKE 5:10,10
453
454 POKE 5:10,10:POKE 5:10,10
455
456 POKE 5:10,10:POKE 5:10,10
457
458 POKE 5:10,10:POKE 5:10,10
459
460 POKE 5:10,10:POKE 5:10,10
461
462 POKE 5:10,10:POKE 5:10,10
463
464 POKE 5:10,10:POKE 5:10,10
465
466 POKE 5:10,10:POKE 5:10,10
467
468 POKE 5:10,10:POKE 5:10,10
469
470 POKE 5:10,10:POKE 5:10,10
471
472 POKE 5:10,10:POKE 5:10,10
473
474 POKE 5:10,10:POKE 5:10,10
475
476 POKE 5:10,10:POKE 5:10,10
477
478 POKE 5:10,10:POKE 5:10,10
479
480 POKE 5:10,10:POKE 5:10,10
481
482 POKE 5:10,10:POKE 5:10,10
483
484 POKE 5:10,10:POKE 5:10,10
485
486 POKE 5:10,10:POKE 5:10,10
487
488 POKE 5:10,10:POKE 5:10,10
489
490 POKE 5:10,10:POKE 5:10,10
491
492 POKE 5:10,10:POKE 5:10,10
493
494 POKE 5:10,10:POKE 5:10,10
495
496 POKE 5:10,10:POKE 5:10,10
497
498 POKE 5:10,10:POKE 5:10,10
499
500 POKE 5:10,10:POKE 5:10,10
501
502 POKE 5:10,10:POKE 5:10,10
503
504 POKE 5:10,10:POKE 5:10,10
505
506 POKE 5:10,10:POKE 5:10,10
507
508 POKE 5:10,10:POKE 5:10,10
509
510 POKE 5:10,10:POKE 5:10,10
511
512 POKE 5:10,10:POKE 5:10,10
513
514 POKE 5:10,10:POKE 5:10,10
515
516 POKE 5:10,10:POKE 5:10,10
517
518 POKE 5:10,10:POKE 5:10,10
519
520 POKE 5:10,10:POKE 5:10,10
521
522 POKE 5:10,10:POKE 5:10,10
523
524 POKE 5:10,10:POKE 5:10,10
525
526 POKE 5:10,10:POKE 5:10,10
527
528 POKE 5:10,10:POKE 5:10,10
529
530 POKE 5:10,10:POKE 5:10,10
531
532 POKE 5:10,10:POKE 5:10,10
533
534 POKE 5:10,10:POKE 5:10,10
535
536 POKE 5:10,10:POKE 5:10,10
537
538 POKE 5:10,10:POKE 5:10,10
539
540 POKE 5:10,10:POKE 5:10,10
541
542 POKE 5:10,10:POKE 5:10,10
543
544 POKE 5:10,10:POKE 5:10,10
545
546 POKE 5:10,10:POKE 5:10,10
547
548 POKE 5:10,10:POKE 5:10,10
549
550 POKE 5:10,10:POKE 5:10,10
551
552 POKE 5:10,10:POKE 5:10,10
553
554 POKE 5:10,10:POKE 5:10,10
555
556 POKE 5:10,10:POKE 5:10,10
557
558 POKE 5:10,10:POKE 5:10,10
559
560 POKE 5:10,10:POKE 5:10,10
561
562 POKE 5:10,10:POKE 5:10,10
563
564 POKE 5:10,10:POKE 5:10,10
565
566 POKE 5:10,10:POKE 5:10,10
567
568 POKE 5:10,10:POKE 5:10,10
569
570 POKE 5:10,10:POKE 5:10,10
571
572 POKE 5:10,10:POKE 5:10,10
573
574 POKE 5:10,10:POKE 5:10,10
575
576 POKE 5:10,10:POKE 5:10,10
577
578 POKE 5:10,10:POKE 5:10,10
579
580 POKE 5:10,10:POKE 5:10,10
581
582 POKE 5:10,10:POKE 5:10,10
583
584 POKE 5:10,10:POKE 5:10,10
585
586 POKE 5:10,10:POKE 5:10,10
587
588 POKE 5:10,10:POKE 5:10,10
589
590 POKE 5:10,10:POKE 5:10,10
591
592 POKE 5:10,10:POKE 5:10,10
593
594 POKE 5:10,10:POKE 5:10,10
595
596 POKE 5:10,10:POKE 5:10,10
597
598 POKE 5:10,10:POKE 5:10,10
599
600 POKE 5:10,10:POKE 5:10,10
601
602 POKE 5:10,10:POKE 5:10,10
603
604 POKE 5:10,10:POKE 5:10,10
605
606 POKE 5:10,10:POKE 5:10,10
607
608 POKE 5:10,10:POKE 5:10,10
609
610 POKE 5:10,10:POKE 5:10,10
611
612 POKE 5:10,10:POKE 5:10,10
613
614 POKE 5:10,10:POKE 5:10,10
615
616 POKE 5:10,10:POKE 5:10,10
617
618 POKE 5:10,10:POKE 5:10,10
619
620 POKE 5:10,10:POKE 5:10,10
621
622 POKE 5:10,10:POKE 5:10,10
623
624 POKE 5:10,10:POKE 5:10,10
625
626 POKE 5:10,10:POKE 5:10,10
627
628 POKE 5:10,10:POKE 5:10,10
629
630 POKE 5:10,10:POKE 5:10,10
631
632 POKE 5:10,10:POKE 5:10,10
633
634 POKE 5:10,10:POKE 5:10,10
635
636 POKE 5:10,10:POKE 5:10,10
637
638 POKE 5:10,10:POKE 5:10,10
639
640 POKE 5:10,10:POKE 5:10,10
641
642 POKE 5:10,10:POKE 5:10,10
643
644 POKE 5:10,10:POKE 5:10,10
645
646 POKE 5:10,10:POKE 5:10,10
647
648 POKE 5:10,10:POKE 5:10,10
649
650 POKE 5:10,10:POKE 5:10,10
651
652 POKE 5:10,10:POKE 5:10,10
653
654 POKE 5:10,10:POKE 5:10,10
655
656 POKE 5:10,10:POKE 5:10,10
657
658 POKE 5:10,10:POKE 5:10,10
659
660 POKE 5:10,10:POKE 5:10,10
661
662 POKE 5:10,10:POKE 5:10,10
663
664 POKE 5:10,10:POKE 5:10,10
665
666 POKE 5:10,10:POKE 5:10,10
667
668 POKE 5:10,10:POKE 5:10,10
669
670 POKE 5:10,10:POKE 5:10,10
671
672 POKE 5:10,10:POKE 5:10,10
673
674 POKE 5:10,10:POKE 5:10,10
675
676 POKE 5:10,10:POKE 5:10,10
677
678 POKE 5:10,10:POKE 5:10,10
679
680 POKE 5:10,10:POKE 5:10,10
681
682 POKE 5:10,10:POKE 5:10,10
683
684 POKE 5:10,10:POKE 5:10,10
685
686 POKE 5:10,10:POKE 5:10,10
687
688 POKE 5:10,10:POKE 5:10,10
689
690 POKE 5:10,10:POKE 5:10,10
691
692 POKE 5:10,10:POKE 5:10,10
693
694 POKE 5:10,10:POKE 5:10,10
695
696 POKE 5:10,10:POKE 5:10,10
697
698 POKE 5:10,10:POKE 5:10,10
699
700 POKE 5:10,10:POKE 5:10,10
701
702 POKE 5:10,10:POKE 5:10,10
703
704 POKE 5:10,10:POKE 5:10,10
705
706 POKE 5:10,10:POKE 5:10,10
707
708 POKE 5:10,10:POKE 5:10,10
709
710 POKE 5:10,10:POKE 5:10,10
711
712 POKE 5:10,10:POKE 5:10,10
713
714 POKE 5:10,10:POKE 5:10,10
715
716 POKE 5:10,10:POKE 5:10,10
717
718 POKE 5:10,10:POKE 5:10,10
719
720 POKE 5:10,10:POKE 5:10,10
721
722 POKE 5:10,10:POKE 5:10,10
723
724 POKE 5:10,10:POKE 5:10,10
725
726 POKE 5:10,10:POKE 5:10,10
727
728 POKE 5:10,10:POKE 5:10,10
729
730 POKE 5:10,10:POKE 5:10,10
731
732 POKE 5:10,10:POKE 5:10,10
733
734 POKE 5:10,10:POKE 5:10,10
735
736 POKE 5:10,10:POKE 5:10,10
737
738 POKE 5:10,10:POKE 5:10,10
739
740 POKE 5:10,10:POKE 5:10,10
741
742 POKE 5:10,10:POKE 5:10,10
743
744 POKE 5:10,10:POKE 5:10,10
745
746 POKE 5:10,10:POKE 5:10,10
747
748 POKE 5:10,10:POKE 5:10,10
749
750 POKE 5:10,10:POKE 5:10,10
751
752 POKE 5:10,10:POKE 5:10
```



134166000	2000	003	032	050	139	040	003	003	004	139	179	099
134166002	2007	004	006	040	001	173	004	007	001	150	100	000
134166003	2000	004	006	000	003	001	173	005	007	140	000	011
134166007	0032	150	139	162	003	060	000	000	000	003	010	017
134166008	2000	002	024	156	002	044	156	002	034	137	002	001
134166009	150	002	030	139	000	000	003	000	139	000	005	002
134166010	100	150	002	001	150	003	047	150	139	000	000	000

32780+149	885	141	190	287	149	672	141	100	267	633	619	221
32780+149	887	141	190	287	149	672	141	100	267	633	619	221
32793+112	141	189	267	149	669	141	669	268	624	167	167	262
32800+000	262	262	234	233	200	240	149	671	141	254	207	178
32810+149	126	141	252	267	149	262	140	645	132	140	162	665
32820+167	149	126	141	608	212	169	664	140	134	667	149	686
32840+000	141	246	267	149	671	132	216	265	149	141	126	136
32850+132	268	173	614	126	641	254	140	614	230	145	601	141
32860+041	256	123	661	146	650	165	660	256	132	660	660	678
32870+085	660	256	123	660	661	660	660	218	132	660	662	661
32880+085	660	211	132	660	660	165	660	212	132	660	664	664
32890+085	660	212	132	660	660	165	660	214	132	660	664	664
32912+105	660	212	132	660	667	266	266	665	665	666	669	636
32920+084	132	661	132	614	126	669	661	141	614	126	149	211
32930+084	140	660	132	149	660	141	624	666	132	183	120	662
32940+074	214	214	126	149	137	141	612	120	149	141	146	146
32950+036	268	669	660	141	618	268	173	617	268	641	127	246
32970+141	617	268	167	123	119	141	620	663	149	141	121	212
32980+080	666	676	632	129	120	674	249	120	149	669	123	176
32990+132	146	660	132	126	149	660	145	211	146	149	211	268
33000+230	252	664	123	126	120	268	245	676	149	616	165	668
33010+080	152	666	664	152	666	665	152	666	156	150	660	626
33020+067	266	266	269	149	621	141	249	667	149	666	162	142
33040+180	214	123	661	147	132	160	214	132	660	219	260	612
33050+266	141	632	162	137	632	643	129	634	167	129	632	166
33060+132	179	660	130	661	149	660	141	265	267	172	678	678
33080+250	267	142	660	662	279	129	632	234	131	240	667	141
33090+136	172	630	268	246	172	250	267	612	149	620	141	166
33100+250	267	141	259	144	232	149	614	141	232	267	174	226
33110+250	267	149	259	652	279	129	636	232	224	131	268	227
33120+267	172	255	267	624	132	626	141	265	267	141	131	642
33140+144	226	169	141	141	160	267	172	265	267	142	166	166
33150+632	259	174	226	124	151	266	665	266	192	266	268	142
33160+242	172	267	267	636	132	626	141	266	267	261	623	166
33170+174	225	149	626	141	265	267	174	265	267	140	120	122
33180+632	259	174	226	123	134	136						



133500.041	181	208	061	076	206	777	287	184	194	811	208	062	
133501.143	174	189	267	028	182	157	840	064	176	212	145		
133502.144	173	282	267	018	148	207	176	080	208	131			
133503.141	206	207	076	011	131	167	001	141	264	267	173	131	
133504.207	267	028	089	013	141	014	208	175	264	108	086		
133505.000	034	136	106	140	206	207	173	016	208	241	127	099	
133506.012	206	207	141	016	206	175	240	267	018	042	042	018	
133507.041	013	013	208	149	000	144	046	208	018	023	141	023	029
133508.067	162	254	154	173	055	208	141	265	267	149	159	058	
133509.141	021	008	032	080	143	021	080	143	021	080	143	021	080
133510.208	008	141	262	267	149	041	240	267	173	202	11		
133511.267	074	144	088	149	010	141	204	267	076	143	131	044	
133512.049	009	141	264	267	173	203	267	141	243	267	149	046	
133513.000	140	184	267	052	262	058	173	262	267	174	208	072	
133514.267	264	240	267	208	008	234	240	267	208	063	076	129	
133515.208	131	052	052	134	206	001	238	206	267	173	074	072	
133516.267	029	159	020	203	140	240	018	241	130	240	066	184	
133517.041	208	207	076	112	131	208	202	267	173	203	267	173	
133518.173	028	208	064	149	000	081	150	132	032	218	250	143	
133519.266	143	021	208	068	149	080	174	262	267	052	068	208	
133520.189	149	052	032	218	250	149	042	012	230	250	149	074	
133521.052	032	218	250	149	000	149	267	032	208	089	029		
133522.067	062	052	218	250	149	061	032	218	250	149	052	062	
133523.266	218	250	173	242	267	141	240	267	173	242	267	173	
133524.141	264	267	072	262	132	174	242	267	173	242	267	173	
133525.022	268	149	067	052	052	218	250	149	044	052	218	052	
133526.233	173	242	267	208	079	240	084	024	085	141	011		
133527.000	267	173	242	267	056	233	056	141	242	267	140	042	
133528.267	267	173	242	267	055	200	041	240	267	053	221	055	
133529.200	221	029	044	018	142	255	160	000	208	253	129		
133530.212	208	250	142	011	052	055	134	061	034	132	234	072	
133531.269	008	141	021	208	146	076	132	040	049	032	132	040	
133532.134	140	000	133	251	133	251	133	040	251	133	040	054	
133533.208	249	139	262	238	234	166	262	224	127	208	139	183	
133534.177	267	143	251	208	172	044	208	247	047	143	143	122	
133535.044	267	173	254	062	141	208	062	147	147	143	052	201	









Nick Hampshire brings  
you some more  
commands to  
improve your Basic.

# BUILD A BETTER BASIC

IN THE LAST FOUR ARTICLES in this series I have given all the initialisation and wedge routines needed to add extra commands to the Basic of a C64 computer. I have also given the code to add 12 new commands which are: CTL, APPEND, CHANGE, DUMP, FIND, AUTO, CHAIN, DELETE, RENUMBER, MARK, SORT and SAMPLE.

This month I am adding a further eight commands. These are: CATALOG, DATA, EXEC, MERGE, GET, PUT, TYPE and OLD. Seven of the eight are special-disk control commands, and eight (OLD) is included since it is called by the other routines. These disk control commands add some very powerful and useful features to a disk-based 64 system and will save a considerable amount of programming time.

All eight new commands require that the wedge and initialisation code given in the first issue are present in memory at the correct locations, and that their command names and entry points are stored in the correct tables. These eight commands are independent of all the previously added commands - except APPEND - routines from which are required by the new routines. Within this limitation they can be used without the previously added code. To ensure that you have the wedges and new routines correctly positioned, the Basic loader at the end of this article gives the initialisation routines and all commands.

In next month's issue I will show how to write and add your own commands to Basic. All the programs used in this series are extracted from the book *Advanced Commodore 64 Basic Revealed* by Nick Hampshire and published by Collins.

## GET

**Associated entry:** CATALOG  
**Affected Basic abbreviations:** None  
**Notes:** Hex SET, EOF, Decimal 28,35

**Mode:** Direct and program  
**Recommended modes:** Direct; differing effects in direct mode and program mode

**Purpose:** To input an ASCII file on disk into memory with line numbers created from 1000 in steps of 10. GET will read in files created by the Commodore assembler and SYNTAX. Each time a read is made a carriage return is reached. It is then tabulated and entered into memory as a program line.  
**Syntax:** Direct mode: GET filename, d - where d is the device number (disk only)

Run mode: as chapter 7:GET and GET #.

**Errors:** Illegal device - if the device number specified is less than eight. Missing file - missing file name - if a null filename is specified.

File not found - if the file does not exist.

Device not present - connected  
File open error - if 10 files are already open

Disk errors - at the end, the disk error channel is read and displayed.

**Uses for editing Commodore assembler files or files for the use of the BASIC command**

**Baseline entry point:** 180171

**Baseline operation:** The GET routine first checks whether the computer is in run mode or direct. If it is in run mode, then the Basic version of GET is performed. If in direct mode, the file parameters are read in and

checked for a null filename or the device not being disk. If these checks are OK, the message 'reading' filename is displayed and the file is opened. Each line is then input and stored in the input buffer, tabulated, and entered into memory until the end of file marker is reached. The program is then re-chained and the variable pointers are set to the correct values for the program. Finally the disk error channel is read and displayed.

## GET

```

1000 GET LIN #FF; (LOWER OF DIRECT)
1010 END RETURN (NOT DIRECT)
1020 FOR NEXT GET CURRENT CHN
1030 ON BASED (POSITION BASIC 'GET'
1040 GETLIN FOR SPACE (GET FILE PARAMETER)
1050 FOR GETLIN ('HEADLINE')
1060 FOR GETLIN (OPEN FILE
1070 FOR GETLIN (SET INPUT
1080 FOR GET (SET START OF PROGRAM
1090 GET # (POSITION)
1100 FOR GET
1110 GET #
1120 FOR GET
1130 GET
1140 GET #
1150 GET
1160 GET #
1170 GET #
1180 GET #
1190 GET #
1200 GET #
1210 GET #
1220 GET #
1230 GET #
1240 GET # (GET LINE #
1250 GET # (GET LINE #
1260 GET # (GET #
1270 GET #
1280 GETLIN FOR #
1290 GETLIN FOR # (GET #
1300 ON BASED (GET OF LINE)

```

```

1010 BSC SETLN :YES
1020 CPM HNDL :LINE POINT
1030 BSC SETLPC :YES
1040 STA MCHNL,Y :STORE BYTE
1050 CPM
1060 CPM HNDL :END OF SUPPORT
1070 BSC SETLPC
1080 SETLN LSH HNDL :STATUS
1090 STA SETLN
1100 LSH HNDL :TERMINATOR
1110 STA MCHNL,Y :STORE
1120 LSH HNDL
1130 STI HNDL
1140 LSH HNDL
1150 STA HNDL
1160 JOP HNDL :COMMON LINE
1170 LSH HNDL
1180 BSC SETLPC :NULL LINE
1190 LSH HNDL
1200 LSH SETLPC :LOWER LB
1210 STA MCHNL,Y :STORE IT
1220 CPM
1230 LSH SETLPC :LOWER H
1240 STA MCHNL,Y :STORE IT
1250 SETLPC CPM
1260 LSH MCHNL,Y :GET BYTE
1270 STA MCHNL,Y :STORE IT
1280 BSC SETLPC :UNTIL END OF LINE
1290 CPM
1300 TPA
1310 LSH HNDL
1320 CLE
1330 BSC MFB :INCREASE POINTER BY
1340 STA MFB :LENGTH
1350 BSC MFB :Y
1360 LSH MFB
1370 BSC HNDL
1380 JOP
1390 STA MCHNL,Y
1400 TPA
1410 LSH MFB
1420 STA MFB
1430 STI MFB
1440 TPA
1450 BSC MFB :END OF PROGRAM
1460 CLE
1470 LSH MFB :INCREASE POINTER BY
1480 STA SETLPC
1490 LSH SETLPC+1
1500 BSC HNDL
1510 STA SETLPC+1
1520 SETLPC LSH SETLN :STATUS
1530 BSC SETLPC :LOAD
1540 JOP SETLPC :GO NEXT LINE
1550 CPM
1560 SETLPC LSH HNDL
1570 TPA
1580 STA MCHNL,Y :END OF PROGRAM
1590 CPM
1600 STA MCHNL,Y
1610 LSH MFB :CLOSE AND DISK
1620 JOP HNDL :RESET POINTER
1630 JOP HNDL
1640 BSC LSH HNDL :POINTER TO
1650 LSH HNDL :INCREASE POINTER TO
1660 LSH HNDL :SET FILE #
1670 JOP HNDL :PRINT STRING
1680 JOP HNDL :PRINT FILENAME
1690 BSC HNDL :SET INPUT
1700 SETLN LSH HNDL
1710 BSC MCHNL :COMMON
1720 CPM
1730 SETLPC LSH HNDL :SET BYTE
1740 STA MCHNL,Y :STORE IT
1750 CPM
1760 CPM HNDL :END OF BIT
1770 BSC SETLPC :SET BIT
1780 LSH HNDL
1790 SETLPC LSH SETLN :SET BYTE
1800 STA MCHNL,Y :STORE IT
1810 CPM
1820 CPM HNDL :END OF BIT
1830 BSC SETLPC :SET BIT
1840 LSH HNDL
1850 STA HNDL
1860 JOP MCHNL :COMPLETE OPEN
1870 SETLN :SET "L,R"
1880 SETLPC :OPEN 0
1890 SETLN :SET 0
1900 :END

```

## CATALOG

### Abbreviation: CATALOG

Affected Basic abbreviations: NONE

Index: Hex \$01505, Decimal 3685

Modes: Direct and program

Recommended mode: Direct  
 Response: To display the directory (CATALOG) of a disk in drive unit (DRIVE). This command will display the directory straight to the screen without having to load it in.

Users of dual-disk drives will be pleased to note that you can specify which drive to display by either a number one or zero after the command. If no number is specified, the routine will default to drive zero.

Syntax: CATALOG [0 or 1]  
 Errors: Syntax error - if the command CATALOG is followed by anything but "0," "1," or nothing.

Disk error message - after the CATALOG has been displayed,

the disk error channel is read and displayed.

Uses: The command is used to display the directory of a disk. This can be useful if you have a program that you wish to save but need to check if there is room on the disk or find a filename to use. The directory can be paused when displaying, by use of the spacebar, and restarted with any key. Display can be stopped completely with the STOP key.

Routine entry point: \$01505

Routine operations: On entry, the routine checks to see if a drive number is specified. If no number is specified or zero, the character "0" is inserted into the filename after the "L". If it is a one, the character "1" is inserted. Anything else will cause system error. The file is then opened and each line is read and displayed ignoring line feeds. When the directory is finished, the file is closed and the disk error channel is read. Check in the one following syll.

## CATALOGUE

```

0000 CATALOG BSC SETLPC :DRIVE 0
0010 CPM HNDL :LS IT BT
0020 BSC SETLPC :YES
0030 CPM HNDL :LS IT BT
0040 BSC SETLPC :YES
0050 JOP MCHNL :FIND COMMON
0060 CATALOG LSH HNDL :CHNL "0"
0070 :SET HNDL
0080 CATALOG LSH HNDL :CHNL "1"

```

```

1090 STA MCHNL,Y :STORE IN STRNG
1100 LSH HNDL :LENGTH
1110 LSH HNDL :ADDRESS LSH
1120 LSH HNDL :END
1130 JOP HNDL :SET FILENAME DETAILS
1140 LSH HNDL
1150 JOP HNDL :SET UNKND FOLN
1160 LSH HNDL :SERVICE 0
1170 LSH HNDL :SA 0
1180 JOP HNDL :SET FILE DETAILS
1190 JOP HNDL :OPEN FILE
1200 BSC SETLPC :NO ERROR

```

```

1210 TPA :STORE ERROR
1220 LSH HNDL :SET FILE #
1230 JOP HNDL :CLOSE FILE
1240 TPA :SET ERROR
1250 JOP MCHNL :END ERROR
1260 CPM
1270 SETLPC LSH HNDL
1280 CATALOG SET HNDL
1290 LSH HNDL
1300 JOP HNDL :SET INPUT SERVICE
1310 JOP HNDL :INPUT
1320 STA HNDL :STORE VALUE

```

## DISK

**Abbreviated entry:** Disk(s)  
**Affected Basic Abbreviations:**  
Disk - Disk

**Token:** Hex \$D5,\$A, Decimal 215,10

**Modem:** Direct and program  
**Recommended modes:** Inter  
**Purpose:** To send a disk command to the disk unit.

**Syntax:** Disk (string expression)  
—where the string expression is:

"S-FILE" - to scratch the file  
ret.

"N-DISK,RET" - to reformat the  
entire disk.

The other syntax is Disk which  
will display the disk error  
message to the screen giving a  
message like:

23, READ ERROR,18,0

where 23 is the error number,  
18 is the track, 01 is the sector,  
and READ ERROR is the error  
description.

**Severe syntax error** - If the first  
character of the command is  
not a quote character.

**String too long** - If the  
command is over 255 bytes  
long.

**Type mismatch** - If the com-  
mand is a number, not a string.

**Use:** This command is useful in  
checking errors caused from  
disk access by using just Disk,  
which displays the message. A  
Basic equivalent would be:

```
OPEN FILE,15
INPUT # 15,NAME,LS
PRINT # 15, "NAME", "L", "S"
CLOSE 15
```

Also, for sending disk  
commands such as scratch a file  
etc.

**Disk "10"**

is equivalent to:

```
OPEN FILE,15,"10"
```

For disk commands, refer to

the disk unit manual.

**Routine entry point:** \$D5-\$D7

**Routine operation:** The Disk  
routine checks to see if  
anything follows the command.  
If not the error channel is read  
and displayed. If there is text  
after the command (which  
must start with the quote  
character) the text is read in  
and sent in the open  
command. Before either of  
these two operations are  
actioned, the current file is  
closed.

```
1150 JSP $PFRF : GET STATUS
1160 BND $D7,17 : STATUS ERROR
1170 JSP $PFRF : INPUT
1180 STA $D7+1 : STORE IT
1190 JSP $PFRF : GET STATUS
1200 BND $D7,16 : STATUS ERROR
1210 LSP $D7 : GET COUNTER
1220 BND $D7,15 : GET KEY
1230 $D7,15 : BND $D7,14
1240 STA $D7 : GET KEY TO $D7
1250 $D7,14 : JSP $PFRF : INPUT
1260 PND : STORE IT
1270 JSP $PFRF : GET STATUS
1280 LND $D7 : STORE TO 0
1290 PLS : GET INPUT CHAR
1300 CPU $D7 : ARE THERE AN ERRORS
1310 BND $D7,13 : YES
1320 LSP $D7 : GET LENGTH
1330 CPU $D7 : TOO LONG?
1340 BND $D7,12 : YES, ERROR
1350 STA $D7,10 : STORE CHARACTER
1360 TND
1370 BND $D7,11 : END OF LINE
1380 LND $D7 : NO NEXT CHAR
1390 JSP $D7,10 : JUMP TO NEXT CHAR
1400 BND $D7,10 : JUMP
1410 JSP $D7,10 : JUMP TO NEXT CHAR
1420 LND $D7 : NO NEXT CHAR
1430 JSP $D7,10 : JUMP TO NEXT CHAR
1440 LND $D7 : NO NEXT CHAR
1450 JSP $D7,10 : JUMP TO NEXT CHAR
1460 LND $D7 : NO NEXT CHAR
1470 JSP $D7,10 : JUMP TO NEXT CHAR
1480 LND $D7 : NO NEXT CHAR
1490 JSP $D7,10 : JUMP TO NEXT CHAR
1500 LND $D7 : NO NEXT CHAR
1510 JSP $D7,10 : JUMP TO NEXT CHAR
1520 LND $D7 : NO NEXT CHAR
1530 JSP $D7,10 : JUMP TO NEXT CHAR
1540 LND $D7 : NO NEXT CHAR
1550 JSP $D7,10 : JUMP TO NEXT CHAR
1560 LND $D7 : NO NEXT CHAR
1570 JSP $D7,10 : JUMP TO NEXT CHAR
1580 LND $D7 : NO NEXT CHAR
1590 JSP $D7,10 : JUMP TO NEXT CHAR
1600 LND $D7 : NO NEXT CHAR
1610 JSP $D7,10 : JUMP TO NEXT CHAR
1620 LND $D7 : NO NEXT CHAR
1630 JSP $D7,10 : JUMP TO NEXT CHAR
1640 LND $D7 : NO NEXT CHAR
1650 JSP $D7,10 : JUMP TO NEXT CHAR
1660 LND $D7 : NO NEXT CHAR
1670 JSP $D7,10 : JUMP TO NEXT CHAR
1680 LND $D7 : NO NEXT CHAR
1690 JSP $D7,10 : JUMP TO NEXT CHAR
1700 LND $D7 : NO NEXT CHAR
1710 JSP $D7,10 : JUMP TO NEXT CHAR
1720 LND $D7 : NO NEXT CHAR
1730 JSP $D7,10 : JUMP TO NEXT CHAR
1740 LND $D7 : NO NEXT CHAR
1750 JSP $D7,10 : JUMP TO NEXT CHAR
1760 LND $D7 : NO NEXT CHAR
1770 JSP $D7,10 : JUMP TO NEXT CHAR
1780 LND $D7 : NO NEXT CHAR
1790 JSP $D7,10 : JUMP TO NEXT CHAR
1800 LND $D7 : NO NEXT CHAR
1810 JSP $D7,10 : JUMP TO NEXT CHAR
1820 LND $D7 : NO NEXT CHAR
1830 JSP $D7,10 : JUMP TO NEXT CHAR
1840 LND $D7 : NO NEXT CHAR
1850 JSP $D7,10 : JUMP TO NEXT CHAR
1860 LND $D7 : NO NEXT CHAR
1870 JSP $D7,10 : JUMP TO NEXT CHAR
1880 LND $D7 : NO NEXT CHAR
1890 JSP $D7,10 : JUMP TO NEXT CHAR
1900 LND $D7 : NO NEXT CHAR
1910 JSP $D7,10 : JUMP TO NEXT CHAR
1920 LND $D7 : NO NEXT CHAR
1930 JSP $D7,10 : JUMP TO NEXT CHAR
1940 LND $D7 : NO NEXT CHAR
1950 JSP $D7,10 : JUMP TO NEXT CHAR
1960 LND $D7 : NO NEXT CHAR
1970 JSP $D7,10 : JUMP TO NEXT CHAR
1980 LND $D7 : NO NEXT CHAR
1990 JSP $D7,10 : JUMP TO NEXT CHAR
2000 LND $D7 : NO NEXT CHAR
```

## DISK

```
1000 DISK JSP $D7,1 : CHECK FOR BLANK
1010 BND $D7,1 : AFTER COMMAND.
1020 JSP $D7,1 : JUMP TO NEXT
1030 BND $D7,1 : JUMP TO NEXT
1040 BND $D7,1 : JUMP TO NEXT
1050 BND $D7,1 : JUMP TO NEXT
1060 BND $D7,1 : JUMP TO NEXT
1070 BND $D7,1 : JUMP TO NEXT
1080 BND $D7,1 : JUMP TO NEXT
1090 BND $D7,1 : JUMP TO NEXT
1100 BND $D7,1 : JUMP TO NEXT
1110 BND $D7,1 : JUMP TO NEXT
1120 BND $D7,1 : JUMP TO NEXT
1130 BND $D7,1 : JUMP TO NEXT
1140 BND $D7,1 : JUMP TO NEXT
1150 BND $D7,1 : JUMP TO NEXT
1160 BND $D7,1 : JUMP TO NEXT
1170 BND $D7,1 : JUMP TO NEXT
1180 BND $D7,1 : JUMP TO NEXT
1190 BND $D7,1 : JUMP TO NEXT
1200 BND $D7,1 : JUMP TO NEXT
1210 BND $D7,1 : JUMP TO NEXT
1220 BND $D7,1 : JUMP TO NEXT
1230 BND $D7,1 : JUMP TO NEXT
1240 BND $D7,1 : JUMP TO NEXT
1250 BND $D7,1 : JUMP TO NEXT
1260 BND $D7,1 : JUMP TO NEXT
1270 BND $D7,1 : JUMP TO NEXT
1280 BND $D7,1 : JUMP TO NEXT
1290 BND $D7,1 : JUMP TO NEXT
1300 BND $D7,1 : JUMP TO NEXT
1310 BND $D7,1 : JUMP TO NEXT
1320 BND $D7,1 : JUMP TO NEXT
1330 BND $D7,1 : JUMP TO NEXT
1340 BND $D7,1 : JUMP TO NEXT
1350 BND $D7,1 : JUMP TO NEXT
1360 BND $D7,1 : JUMP TO NEXT
1370 BND $D7,1 : JUMP TO NEXT
1380 BND $D7,1 : JUMP TO NEXT
1390 BND $D7,1 : JUMP TO NEXT
1400 BND $D7,1 : JUMP TO NEXT
1410 BND $D7,1 : JUMP TO NEXT
1420 BND $D7,1 : JUMP TO NEXT
1430 BND $D7,1 : JUMP TO NEXT
1440 BND $D7,1 : JUMP TO NEXT
1450 BND $D7,1 : JUMP TO NEXT
1460 BND $D7,1 : JUMP TO NEXT
1470 BND $D7,1 : JUMP TO NEXT
1480 BND $D7,1 : JUMP TO NEXT
1490 BND $D7,1 : JUMP TO NEXT
1500 BND $D7,1 : JUMP TO NEXT
1510 BND $D7,1 : JUMP TO NEXT
1520 BND $D7,1 : JUMP TO NEXT
1530 BND $D7,1 : JUMP TO NEXT
1540 BND $D7,1 : JUMP TO NEXT
1550 BND $D7,1 : JUMP TO NEXT
1560 BND $D7,1 : JUMP TO NEXT
1570 BND $D7,1 : JUMP TO NEXT
1580 BND $D7,1 : JUMP TO NEXT
1590 BND $D7,1 : JUMP TO NEXT
1600 BND $D7,1 : JUMP TO NEXT
1610 BND $D7,1 : JUMP TO NEXT
1620 BND $D7,1 : JUMP TO NEXT
1630 BND $D7,1 : JUMP TO NEXT
1640 BND $D7,1 : JUMP TO NEXT
1650 BND $D7,1 : JUMP TO NEXT
1660 BND $D7,1 : JUMP TO NEXT
1670 BND $D7,1 : JUMP TO NEXT
1680 BND $D7,1 : JUMP TO NEXT
1690 BND $D7,1 : JUMP TO NEXT
1700 BND $D7,1 : JUMP TO NEXT
1710 BND $D7,1 : JUMP TO NEXT
1720 BND $D7,1 : JUMP TO NEXT
1730 BND $D7,1 : JUMP TO NEXT
1740 BND $D7,1 : JUMP TO NEXT
1750 BND $D7,1 : JUMP TO NEXT
1760 BND $D7,1 : JUMP TO NEXT
1770 BND $D7,1 : JUMP TO NEXT
1780 BND $D7,1 : JUMP TO NEXT
1790 BND $D7,1 : JUMP TO NEXT
1800 BND $D7,1 : JUMP TO NEXT
1810 BND $D7,1 : JUMP TO NEXT
1820 BND $D7,1 : JUMP TO NEXT
1830 BND $D7,1 : JUMP TO NEXT
1840 BND $D7,1 : JUMP TO NEXT
1850 BND $D7,1 : JUMP TO NEXT
1860 BND $D7,1 : JUMP TO NEXT
1870 BND $D7,1 : JUMP TO NEXT
1880 BND $D7,1 : JUMP TO NEXT
1890 BND $D7,1 : JUMP TO NEXT
1900 BND $D7,1 : JUMP TO NEXT
1910 BND $D7,1 : JUMP TO NEXT
1920 BND $D7,1 : JUMP TO NEXT
1930 BND $D7,1 : JUMP TO NEXT
1940 BND $D7,1 : JUMP TO NEXT
1950 BND $D7,1 : JUMP TO NEXT
1960 BND $D7,1 : JUMP TO NEXT
1970 BND $D7,1 : JUMP TO NEXT
1980 BND $D7,1 : JUMP TO NEXT
1990 BND $D7,1 : JUMP TO NEXT
2000 BND $D7,1 : JUMP TO NEXT
```

```

1400 END GETN2
1410 GETN4 N1
1500 :
1510 SUBROUTINE CMP #0001 : CHECK FOR COMMAND
1520 BND #0000 : ON SCREEN
1530 JMP #0000 : SCREEN GRAPH
1540 BNDN1 LDA #00 : CLOSE CURRENT
1550 STA #00 : END FILE
1560 JMP #0000
1570 JMP #0000 : GET TEXT IN BUFFER
1580 JMP #0000

```

## EXEC

```

1000 EXEC JOB #0000 :GET FILE PARAMETERS
1010 JMP GETN000 :OPEN FILE
1020 LDA #000 :CLEAR SCREEN
1030 JMP #0000
1040 LDA #0000 :STORE OFF SCREEN LINE
1050 STA #0000
1060 LDA #0000
1070 STA #0000+1
1080 LDA #0000 :STORE OFF WARM START
1090 STA #0000
1100 LDA #0000
1110 STA #0000+1
1120 LDA #0000000 :SET "RESET INPUT"
1130 STA #0000 : TO RTS
1140 LDA #0000000
1150 STA #0000
1160 LDA #0000000 :SET ERROR VECTOR"
1170 STA #0000
1180 LDA #0000000
1190 STA #0000
1200 LDA #0000000 :SET WARM START
1210 STA #0000
1220 LDA #0000000
1230 STA #0000
1240 EXEC000 LDA #0000
1250 JMP #0000 :GET INPUT
1260 LDA #0 :RETURN
1270 LDA #000 : LEFT
1280 END
1290 JMP #0000 : ON SCREEN
1300 LDA #000
1310 EXEC000 JMP #000 :SET BYTE
1320 END
1330 PLA
1340 CMP #000 :CARRIAGE RETURN"
1350 BND EXEC000
1360 PLA
1370 CMP #000 :CARRIAGE RETURN"
1380 BND EXEC000
1390 STA #0000,1
1400 END
1410 LDA #000 : STORE ADDRESS AT
1420 STA #000 : (X00)
1430 LDA #000
1440 STA #000
1450 RESET STA #00 : SET LENGTH
1460 JMP #0000 : OPEN FILE
1470 LDA #000
1480 JMP #0000 : PRINT RETURN"
1490 RTS : EXIT HERE
1500 END

```

## EXEC

**Abbreviated entry:** *Exhibit*  
**Affected Basic instructions:**  
 LOP = LOP  
 Volume: *Hex 000000, Decimal 200.00*

**Modem:** Direct and program  
**Recommended reader:** Direct only

**Purpose:** To EXECute a test file stored on disk. This command works in conjunction with GET and PUT.

**Syntax:** EXEC filename.d - where d is the device number (disk only).

**Notes:** Illegal device = 0 the device number specified is less than eight.

**Missing filename** - If a null filename is specified file not found - if the file does not exist.

**Device not present** - if no disk drive is connected  
 Too many files - if 10 files are already open

**Disk error** - at the end, the disk error channel is read and displayed.

User EXEC can be used in several different ways. The main one is to set up function keys when first powered up, for example enter the program:

```

10 CTRL,5,0,1,1
20 KEY1,"CATALOG"+CHR$(10)
30 KEY2,"EXEC"+CHR$(10)
40 KEY3,"LEFT"+CHR$(10)
50 KEY4,"RIGHT"+CHR$(10)
60 KEY5,"CLEAR"+CHR$(10)
70 KEY6,"FMSK"
80 KEY7,"SAVE NUMBER"
90 KEY8,"END"
100 PRINT CTRL(1,1),1,1
"FUNCTION KEYS DEFINED"

```

Use the PUT command to write this to a disk file PUT "FS" 0

When powered up, type EXEC "FS" 0 and the commands will be carried out and your function keys will be defined.

Other uses could be setting of change commands to a program.

**Baseline entry points:** [EXEC]

**Baseline operations:** The filename and device number are read in and the file is opened. Each line is read into the input buffer until carriage return is found. It is then tokenized, and executed until the file is complete or an operating error occurs.

## MERGE

**Abbreviated entry:** MERGE  
**Affected Basic abbreviations:** None

**Tokens:** Hex \$01572, Decimal 23878

**Modes:** Direct and program  
**Recommended mode:** Direct only

**Purpose:** To merge a Basic program from disk into the current Basic program in memory.

**System:** MERGE filename - where n is the device number (disk only).

**Errors:** Illegal device - if the device number specified is less than eight.

**Missing filename:** - if a null filename is specified.

**File not found:** - if the file does not exist.

**Device not present:** - if no disk drive is connected.

**File open error:** - if 10 files are already open.

**Disk errors:** - at the end, the disk error channel is read and displayed.

**Use:** Merge is used to combine two Basic programs in memory. Each line of the program on disk is read in and the zero byte is reached, and stored in the input buffer. The Basic routine to enter a line is then called and the line is entered at the next place. Note: The line number of the program to MERGE is the same as its

existing line number, the associated line will replace it.

**Routine entry point:** \$0700  
**Real-time operations:** The filename and device are read in and checked for missing filename and illegal device. If both checks are OK, the file is opened and the message MERGE23878 is displayed. Each line is then read into the input buffer and entered using the Basic routine to do so. When the file is completed, it is closed and the disk error channel is read and displayed.

### MERGE

```

0000 MERGE J20 SPARE ; SET FILE PARAMETER
0010 LDA #000000 ; DISPLAY MERGE MESSAGE
0020 LSR #000000
0030 J20 MERGE
0040 J20 #FFC1 ; DISPLAY FILENAME
0050 LDA #0000 ; GIVE BASIC WORK START
0060 STA #0000 ; LINE
0070 LDA #0000
0080 STA #0000+1
0090 LDA #0000 ; FIND FILE NUMBER
0100 J20 #2700
0110 STA #0000
0120 J20 #2700
0130 LDA #0000
0140 LDA #0000+1
0150 LDA #0000 ; FIND FILE NUMBER
0160 J20 #2700
0170 J20 #FFC1 ; OPEN FILE
0180 LDA #0000
0190 J20 #FFC1 ; SET FILE TO INPUT
0200 LDA #0000+1
0210 J20 #2700
0220 LDA #0000 ; SET 'INPUT' INPUT
0230 STA #0000 ; TO A #20
0240 LDA #0000+1
0250 STA #0000
0260 LDA #0000+1
0270 J20 #2700
0280 J20 #FFC1 ; INPUT NEXT LINE
0290 J20 #FFC1 ; INPUT NEXT LINE
0300 J20 #FFC1 ; INPUT NEXT LINE
0310 J20 #FFC1 ; INPUT NEXT LINE
0320 J20 #FFC1 ; INPUT NEXT LINE
0330 J20 #FFC1 ; INPUT NEXT LINE
0340 J20 #FFC1 ; INPUT NEXT LINE
0350 J20 #FFC1 ; INPUT NEXT LINE
0360 J20 #FFC1 ; INPUT NEXT LINE
0370 J20 #FFC1 ; INPUT NEXT LINE
0380 J20 #FFC1 ; INPUT NEXT LINE
0390 J20 #FFC1 ; INPUT NEXT LINE
0400 J20 #FFC1 ; INPUT NEXT LINE
0410 J20 #FFC1 ; INPUT NEXT LINE
0420 J20 #FFC1 ; INPUT NEXT LINE
0430 J20 #FFC1 ; INPUT NEXT LINE
0440 J20 #FFC1 ; INPUT NEXT LINE
0450 J20 #FFC1 ; INPUT NEXT LINE
0460 J20 #FFC1 ; INPUT NEXT LINE
0470 J20 #FFC1 ; INPUT NEXT LINE
0480 J20 #FFC1 ; INPUT NEXT LINE
0490 J20 #FFC1 ; INPUT NEXT LINE
0500 J20 #FFC1 ; INPUT NEXT LINE
0510 J20 #FFC1 ; INPUT NEXT LINE
0520 J20 #FFC1 ; INPUT NEXT LINE
0530 J20 #FFC1 ; INPUT NEXT LINE
0540 J20 #FFC1 ; INPUT NEXT LINE
0550 J20 #FFC1 ; INPUT NEXT LINE
0560 J20 #FFC1 ; INPUT NEXT LINE
0570 J20 #FFC1 ; INPUT NEXT LINE
0580 J20 #FFC1 ; INPUT NEXT LINE
0590 J20 #FFC1 ; INPUT NEXT LINE
0600 J20 #FFC1 ; INPUT NEXT LINE
0610 J20 #FFC1 ; INPUT NEXT LINE
0620 J20 #FFC1 ; INPUT NEXT LINE
0630 J20 #FFC1 ; INPUT NEXT LINE
0640 J20 #FFC1 ; INPUT NEXT LINE
0650 J20 #FFC1 ; INPUT NEXT LINE
0660 J20 #FFC1 ; INPUT NEXT LINE
0670 J20 #FFC1 ; INPUT NEXT LINE
0680 J20 #FFC1 ; INPUT NEXT LINE
0690 J20 #FFC1 ; INPUT NEXT LINE
0700 J20 #FFC1 ; INPUT NEXT LINE
0710 J20 #FFC1 ; INPUT NEXT LINE
0720 J20 #FFC1 ; INPUT NEXT LINE
0730 J20 #FFC1 ; INPUT NEXT LINE
0740 J20 #FFC1 ; INPUT NEXT LINE
0750 J20 #FFC1 ; INPUT NEXT LINE
0760 J20 #FFC1 ; INPUT NEXT LINE
0770 J20 #FFC1 ; INPUT NEXT LINE
0780 J20 #FFC1 ; INPUT NEXT LINE
0790 J20 #FFC1 ; INPUT NEXT LINE
0800 J20 #FFC1 ; INPUT NEXT LINE
0810 J20 #FFC1 ; INPUT NEXT LINE
0820 J20 #FFC1 ; INPUT NEXT LINE
0830 J20 #FFC1 ; INPUT NEXT LINE
0840 J20 #FFC1 ; INPUT NEXT LINE
0850 J20 #FFC1 ; INPUT NEXT LINE
0860 J20 #FFC1 ; INPUT NEXT LINE
0870 J20 #FFC1 ; INPUT NEXT LINE
0880 J20 #FFC1 ; INPUT NEXT LINE
0890 J20 #FFC1 ; INPUT NEXT LINE
0900 J20 #FFC1 ; INPUT NEXT LINE
0910 J20 #FFC1 ; INPUT NEXT LINE
0920 J20 #FFC1 ; INPUT NEXT LINE
0930 J20 #FFC1 ; INPUT NEXT LINE
0940 J20 #FFC1 ; INPUT NEXT LINE
0950 J20 #FFC1 ; INPUT NEXT LINE
0960 J20 #FFC1 ; INPUT NEXT LINE
0970 J20 #FFC1 ; INPUT NEXT LINE
0980 J20 #FFC1 ; INPUT NEXT LINE
0990 J20 #FFC1 ; INPUT NEXT LINE

```

```

1000 J20 #0000
1010 J20 #0000
1020 J20 #0000
1030 J20 #0000 ; END OF LINE? NO.
1040 J20 #0000
1050 J20 #0000
1060 J20 #0000
1070 J20 #0000
1080 J20 #0000
1090 J20 #0000
1100 J20 #0000
1110 J20 #0000
1120 J20 #0000
1130 J20 #0000
1140 J20 #0000
1150 J20 #0000
1160 J20 #0000
1170 J20 #0000
1180 J20 #0000
1190 J20 #0000
1200 J20 #0000
1210 J20 #0000
1220 J20 #0000
1230 J20 #0000
1240 J20 #0000
1250 J20 #0000
1260 J20 #0000
1270 J20 #0000
1280 J20 #0000
1290 J20 #0000
1300 J20 #0000
1310 J20 #0000
1320 J20 #0000
1330 J20 #0000
1340 J20 #0000
1350 J20 #0000
1360 J20 #0000
1370 J20 #0000
1380 J20 #0000
1390 J20 #0000
1400 J20 #0000
1410 J20 #0000
1420 J20 #0000
1430 J20 #0000
1440 J20 #0000
1450 J20 #0000
1460 J20 #0000
1470 J20 #0000
1480 J20 #0000
1490 J20 #0000
1500 J20 #0000
1510 J20 #0000
1520 J20 #0000
1530 J20 #0000
1540 J20 #0000
1550 J20 #0000
1560 J20 #0000
1570 J20 #0000
1580 J20 #0000
1590 J20 #0000
1600 J20 #0000
1610 J20 #0000
1620 J20 #0000
1630 J20 #0000
1640 J20 #0000
1650 J20 #0000
1660 J20 #0000
1670 J20 #0000
1680 J20 #0000
1690 J20 #0000
1700 J20 #0000
1710 J20 #0000
1720 J20 #0000
1730 J20 #0000
1740 J20 #0000
1750 J20 #0000
1760 J20 #0000
1770 J20 #0000
1780 J20 #0000
1790 J20 #0000
1800 J20 #0000
1810 J20 #0000
1820 J20 #0000
1830 J20 #0000
1840 J20 #0000
1850 J20 #0000
1860 J20 #0000
1870 J20 #0000
1880 J20 #0000
1890 J20 #0000
1900 J20 #0000
1910 J20 #0000
1920 J20 #0000
1930 J20 #0000
1940 J20 #0000
1950 J20 #0000
1960 J20 #0000
1970 J20 #0000
1980 J20 #0000
1990 J20 #0000

```

## OLD

**Abbreviated entry:** OLD  
**Affected Basic abbreviations:** None

**Tokens:** Hex \$01573, Decimal 23879

**Modes:** Direct and program  
**Recommended mode:** Direct only (there should be no program in memory)

**Purpose:** To restore a Basic program after a NEW has been performed.

**System:** OLD

**Errors:** None

**Use:** OLD can be used if the program in memory has been wiped out using the NEW command. OLD will not work if DELITE was used to remove the whole program or if a variable has been declared since the NEW. In most cases, syntax error will create a variable e.g. U instead of U=0 will create the variable U and give syntax error instead of trying to load the program.

**Routine entry point:** \$0700  
**Real-time operations:** The first line is scanned until the end and the pointer to the next line is returned. The program is then re-compiled and variable pointers are set.

## PUT

**Abbreviated entry:** PUT  
**Affected Basic abbreviations:** None

**Tokens:** Hex \$01574, Decimal 23880

**Modes:** Direct and program  
**Recommended mode:** Direct only

**Purpose:** To list a Basic program to a disk file without line numbers.

**System:** PUT filename - where n is the device number (disk only).

**Errors:** Illegal device - if the device number specified is less than eight.

**Missing filename:** - if a null filename is specified.







# **SJB DISKS LIMITED**

## **BLANK DISKS**

Top Quality 5¼" Bulk Packed Disks

Supplied in 10's with a **FREE** Plastic Library Case, Labels etc.

Single Sided  
(40/80 Track)

Double Sided  
(40/80 Track)

10 – 5¼" Disks  
(With a **FREE** Plastic  
Library Case) .....

**£13.95    £17.95**

50 – 5¼" Disks  
(With a **FREE** Perspex  
Storage Box) .....

**£59.95    £79.95**

★ **LIFETIME WARRANTY** ★

★ **FREE FAST DELIVERY** ★

All prices are inclusive of V.A.T.  
Delivery ...**FREE** throughout the U.K.

3" & 3½" Disks also available

Bulk Orders, Trade Enquiries & Educational Enquiries Welcome



Please Send Cheques/Postal Orders to:-

**SJB DISKS LIMITED (Dept YC )**

**11 Oundle Drive, Nottingham, NG8 1BN**  
**Telephone (0602) 782310**



Garry Marshall  
explains how a  
computer can be used  
to interpret images.

# PROGRAMMING PROJECTS

IN MANY AREAS, COMPUTERS are used to help interpret images such as that in Figure 1. The illustration shows a part of the sky as seen from a powerful telescope, and computers are used to process pictures of this kind to make sense of them in terms of the galaxies and clusters of objects appearing in them. A similar process takes place in giving a robot the capability to see. To be able to recognize the items that it is to assemble, for instance, an industrial robot must be able to interpret the pattern of darker and lighter dots representing its field of vision as presented by a video camera. In both cases, the computer is running a program that enables it to bring something of order to an apparently chaotic scene.

There is another area where computers can be used to help interpret images, and this is in archaeology. Often as a site excavated by archaeologists, the only evidence remaining of a building that once occupied the site is a pattern of holes in the ground. After the building itself has decayed, the holes in which its supporting timbers were placed (known as post-holes) remain. Even for a single building, which will obviously have a rectangular plan, the plan itself is often some two clues. When many buildings have occupied a site at different times with some built over the same ground as earlier, vanished, ones, the overlapping patterns of holes can be chaotic. Computers can be used to good effect in trying to reconstruct the plans of the houses that once occupied a site.

This month's project is to reconstruct the plan of a building from an irregular but more-or-less rectangular pattern of post-holes such as that in Figure 2 by superimposing an outline of the plan on it, as shown in Figure 3.

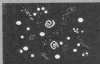


Figure 1 Clues in the skies



Figure 2 A pattern of post-holes

## The Solution

The first thing the program must do is to plot the pattern of post-holes as in Figure 2. This can be done by reading the positions of the post-holes from DATA statements and then plotting a block at each position to represent a post-hole. The positions will be needed again later in finding the plan of the house, so it is worth storing them in arrays as they are read. Using arrays named *XP* and *YP* to hold, respectively, the column and row positions on the screen for the post-holes, and making use of our point-plotting subroutine, which begins with line 3000, the program starts as:

```
10 DIM XPT(5), YPT(5), X(5), Y(5), NP(1)
20 GOSUB 300: REM PREPARE
30 FOR I=1 TO 12
40 READ C, R: XP(I)=C: YP(I)=R
```



Figure 3 Post-holes with rectangular plan superimposed

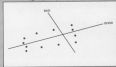


Figure 4 Good and bad approximation to the main axis for a pattern of post-holes

```
50 GOSUB 1800: C=C+1:
60 FOR I=1 TO 5: REM PLOT
70 NEXT I
80 IF NP(1) GOSUB 1000: C=C+1:
90 GOSUB 1800
100 NEXT I
110 DATA 45, 55, 42, 44, 57,
120 50, 105, 105, 105, 115
130 DATA 105, 105, 42, 121, 49,
140 55, 115, 50, 55, 45, 59
```

The next step is to find the main axis of the house, by doing this we shall find the directions of all the walls of the house. The longer walls will be parallel to the axis and the shorter walls must be at right angles to it. Figure 4 shows the idea behind the method for finding the main axis. It shows a line that goes close to the positions of all the post-holes and a second line that obviously goes much further from most of them. The first represents a good approximation to the main axis, and the second a poor one. We shall use a well-

known mathematical method for finding the line passing closest to a set of points, and this will give us the main axis of the house. The method is that of finding the line giving the 'best least squares fit' to a set of points.

We will write the program so that it waits until a key is pressed before going on to calculate the position of the main axis and to display it using our line-drawing subroutine that starts at line 3000. This gives the next section of the program as:

```
180 GET C$: IF C$="" THEN
190 FOR J=1 TO 12: GOTO(YP(J)-
200 YP(1)): NEXT J
210 GOSUB 1800: REM FIND
AND DRAW LINE
```

The subroutine that does all the work is:

```
2600 REM FIND AND DRAW
LINE
3000 L=0: S=0: S2=0: S3=0: S4=0
3100 FOR I=1 TO NP
```



**Romcasters delve into some  
dungeons and dragons games  
and faces danger and death.**

## New and Devious...

GLOBAL IS A RELATIVELY NEW NAME in computer software and it the company's first adventure game - *The Magician's Ball* - is anything to go by it will be worth watching out for in the future. It is by no means perfect as there are several anomalies apparent in playing it... but nevertheless the presentation and the command structure are good and sufficiently different to make it worth your attention.

The program is on cassette only and will run on the C64 and C128. A fast loader is incorporated and the program will load in just over four minutes. The screen display is colourful and the text easily read, with the graphic picture of your location occupying a small rectangle in the top left corner.

The graphics are not particularly exciting but are clear and are 'themed' very simply. Occasional use of sprites adds some movement that gives a little life to the scenes. Unlike some games the pictures, although simple are quite different and clearly give quick visual recognition to one's location.

The main location-description is to the right of the picture and contains plenty of information to build quite a full mental image of your surroundings. Visible rats are indicated and input commands are entered at the bottom of the screen.

The 12 line 'window' between the description and the command line will contain additional information as what is seen at that location - creatures, objects that can be carried and even what other creatures are carrying!

Early in the storyline is that an evil magician has abducted a beautiful princess from her father's side; you - Cam - were there at the time and in a foolish attempt to rescue the girl, got caught away by the magician's magic, as he returned to his own domain. You are now in the magician's realm and... on your own!

Input commands may consist of reasonably complex sentences and also permit you to tell other characters what to do. A very interesting feature of this adventure is that you can swap personalities and 'control' some of the other characters you meet in your travels.

The normal 'game' facilities are attended with 'saving' and 'loading' lines at just under ten minutes but another neat trick from Global is the 'Quicksave' and 'Quickload'. These are implemented by pressing the Commodore key and either 'S' or 'L'.



This 'user' option almost instantaneously stores your present position into a protected area of memory. A very useful command if you think you are about to be killed or even to create a 'fall back' position if you are not sure of your next few moves.

Further 'user friendly' options exist in the input command area - not only can you delete an unwanted letter by using the delete key in the normal manner but you can delete the entire command with SHIFT and CLR.

Pressing 'W' repeats the last command and pressing the left arrow key (top left of the keyboard) recalls the last command for you to modify if desired. I think it is facilities such as these that make Global worth watching in the future. User friendly games are worth cultivating!

The operating system may be user friendly, the game is in many ways just plain devious! All the clues are there but you may have difficulty seeing the wood for the trees. There are several 'with-holding paths' and the number of objects you can carry is not always as great as you may wish - perhaps you should get someone/something to carry them for you!

Throughout your journey, you travel to magical accompaniment from 'Tubular Bells'; this provides a pleasant melody as you set back to ponder your next move.

There are the anomalies I mentioned earlier but these do not alter the game play in any way - they merely appear as

various characters saying something that has no useful or significant purpose!

There are also a number of locations where you will have at one location, two (or more) items that although different are similar - a gold and an ornamental key for instance. It is worth dropping the one you do not want to use at another location as the program sometimes finds it difficult to understand what you want it to do!

*The Magician's Ball* is good fun and both novice and experienced adventurers should find something in it to enjoy.

## Old But Faster

Once upon a time there were no C64s or C128s. I know that is difficult to believe but it's true. There were PETs and Atari and TRS 80s and quite a number of programs for them.

Amongst these there were also a number of adventure games that closely followed the general idea of *Dungeons and Dragons*, with lots of monsters to meet in battle and treasure to be found by the bold and daring.

They taught the imagination of thousands of home computer users in America (the land of their origin) and there were even competitions and conventions where adventurers of the mind could compete against the clock to prove their prowess in the worlds of dungeons and magical trends.

One of the favourites was called *The*

Temple of Apokal. At that time, with the exception of the Amiga, none of the home computers had what we could call hi-res graphics and as it was not written for the Commodore machines in 1989, the conversion for the PC was really lacking.

The graphics showing the rooms and corridors around your character, took an age to be 'drawn' to the screen, the action was slow, the response to most commands (and there were plenty of optional) was painfully slow and last but not least the keyboard buffer was not cleared.

This last 'fault' was charitable because, with the slow response to the input commands (all single key) there was a high probability that you would press one or more keys several times trying to either evade or attack a monster with the consequence that your character would still be sitting at the monster long after it had been slain!

Even a fairly recent conversion for the OS2 still exhibited many of these shortcomings making the game tedious and boring to play. All these half-hearted attempts were really a great shame because the original coverage was worth playing.

I am pleased to report that at long last, a playable version has appeared for the OS2. This is a new (yet another) interpretation entirely, with its superior graphics, faster playing all round, no keyboard buffer problems and to cap it all, entirely the Temple of Apokal but also on the same disc, the two additional adventures that followed: *Upper Reaches of Apokal* and *The Curse of Ra*.

The Temple of Apokal Trilogy is produced by Elys Computer Software (Invisible Monks, Summer Games etc.) and is distributed in the UK by US Gold. One other vital fact is that the three-volume disk will retail for only £5.99.

If you have the slightest inclination towards 'monster bashing', then these programs are for you. Even if you are only faintly interested then the Temple Trilogy is worth considering, purely as being a class of its type.

The instruction manual for the trilogy is some 50 pages long and covers all the basic rules for playing, together with a room by room description of all the various locations. The graphics representation shown on the screen simply depicts a wire of rooms and joining corridors.

It is not absolutely necessary to read the descriptions but it helps set the scene. You will also need to refer to the manual to cross-check against the various monsters that you find.

The three scenarios provided by the Temple Trilogy are in many respects compatible with many non-computer role playing games. You may even 'convert' an existing character in one of these to continue his or her Apokal. When you first start playing you are given several choices.

You may start from scratch, with your

character given a random rating from 1-10 for their six basic attributes: intelligence, intuition, ego, strength, coordination and dexterity, together with a limited amount of silver pieces with which to equip themselves.

You may enter a previously 'saved' character from this or you may enter the statistics of a known character from some other game. Or you may initiate a previous game 'saved' at any point within Apokal.

Your character is 'named' and is your own 'persona' within the confines of the game. If you start from scratch then you will have to buy the various pieces of equipment you think you will need - from the marketplace.

This is an art in itself as you must learn to juggle with how to get the best bargain. Pay the asking price and you will end up with very little equipment!

Choice of equipment is fairly important. Various words are an offer but if you do not have a great deal of strength then a two-handed sword will kill you quickly. A small shield may be very effective if you have high 'dexterity'.

The status screen shown throughout your adventures in Apokal shows wounds and fatigue, the lower percentage of wounds the nearer you are to death. The lower percentage of fatigue the less energy you have to fight or run away.

Wounds may be healed over the passage of time or through the use of 'healing salves' or 'elixirs', the latter may be purchased from the marketplace, the latter can be found within the rooms and corridors of Apokal.

Fatigue may be removed by moving very slowly, resting or by being rescued! The heavier your equipment to treasure, then the more fatigued you will become.

Having beaten the marketplace down in price and purchased a sword, armour, a shield, bow, arrows and a few healing salves, you are ready to enter Apokal. You have the choice of which realm to explore and at what level.

Each realm has four levels and for Dungeons and Dragons followers levels one to two are appropriate to third level characters and the lower levels are more suited to fourth and sixth level characters.

Once you have set out on your present quest you have a number of commands available to you. The latest version of the Trilogy allows for the use of a joystick, with the four directional movements giving you turn 90 degrees left or right, move five feet in the direction you are facing or rest.

The joystick, with the fire button depressed, also offers normal attack, parry, thrust (all out attack) or fire a magical arrow.

Single key commands from the keyboard also allow you to move forward from one to nine feet (keys 1-9), turn to the left, right or turn about (100 degrees), attack, thrust or parry, fire a normal or a

magic arrow.

There are also a number of special commands you may use, such as examining a wall for secret doors, opening doors, travelling by map, picking up treasures, learning for monsters, talking to monsters, healing yourself with salves or elixirs, checking the inventory of your treasures or saving the game or character for future use.

As you may now be beginning to understand, the Temple of Apokal Trilogy is a little more than a straight-forward 'monster bashing' adventure game. A little thought is also required if you are to last any length of time within its dangerous environs.

The option of defining your own character has got to be double edged. It is all too easy to give your character 50 points for all attributes. This strikes too close to, don't you say... cheating?

On the other hand, providing **RUN/STOP HISTORY** if the attributes given to you by the linkkeeper are particularly poor does give you the chance to improve on the random attribute system, after all there are various magical items to be found that will increase some 1 out of all of your vital statistics!

Most levels of each realm have between 30 and 40 different rooms, so there is plenty to explore. Some monsters are 'fixed' to particular locations but there are also plenty that wander around just looking for the next human.

If you choose the right realm and level it is not too difficult to increase your experience and treasure but beware danger and things become a little more difficult.

Should the worst happen, then there is still a chance that all is not lost. If you come to an untimely end one of four things may happen, you may get stolen by a wandering monster... fly-by, or you may be found and resurrected by one of those 'friends of the dragon'.

One will do this for no cost, other than the request of a donation to his holy order, the other two will extract varying degrees of payment. So what? Treasures still allow and able to add to your valuable experience.

If you 'save' your character after each day into the underworld, you should be able to gradually build up a better powerful character. After each return to the inn, the linkkeeper will pay reward for treasure you have collected and you will have the chance of adding or improving your equipment.

Just remember the restrictions that exist before weapons and armour place upon your movement and energy usage.

Also remember that copies of the newer version of Temple of Apokal are still around, make sure you get the latest version. You want to spend weeks over the game, not hours over any 'maver' floppy bashing!





# C16



**Derek Moody gives  
more control to your  
fingers with this article  
for the C16.**

I PURCHASED A C16 AT THE end of March but horror of horrors, there was yet another Micro supplied with an inadequate manual. Commodore suggest that you should buy their Programmer's reference guide, but that doesn't even contain memory map, let alone operating system entry points or connector pinouts. First of all, I needed a decent keyboard control routine, so I disassembled the ROM and started searching for the necessary information. This article is based on some of the results.

The C16 detects keypresses and stores them, even when the computer is doing

something else. This is possible because the keyboard is being read in an interrupt routine, this routine also maintains the real time clock and does a certain amount of house-keeping for the operating system. The 18Q interrupt routine is called 48 times each second, and there are three points at which it is vectored through RAM, at \$112, \$114, and \$116. The vector that will be of most interest to us is at \$112, the computer refers to it after reset of the house-keeping, but before updating the real time clock and reading the keyboard. A vector, by the way, is an address held in two bytes of RAM, which points to a block of machine code in ROM, by altering a vector, the programmer can cause his own block of code to be used instead.

The keyboard is read by a short 18 line routine at \$0570 this works by writing the contents of the accumulator to the columns of the keyboard matrix, and reading the rows

## Program Listing 1

```
10 REM C16 KEYBOARD MATRIX DEMO
11 REM
12 REM BY DEREK MOODY APRIL 1984
13 REM
140 :
40 TOS=3872+41
50 OS=14335
60 COSUB 2000
77 :
90 REM CONTROL ROUTINE
99 :
100 GOSUB 2000
110 GO
120 : GOSUB 1000
130 LOOP
140 END
997 :
995 READ MATRIX, AND PLOT RESULTS
999 :
1000 XXX=1
1010 FOR I=0 TO 7
1020 : POKE XX,XXX
```



# Program Listing 1 (cont.)

```

1838 : SYS 10X+11
1848 : R3=PEEK(10X)
1858 : Y3=1
1868 : FOR Y=0 TO 7
1878 :   C3=C2
1888 :   IF (R3 AND Y3) THEN C3=160
1898 :   POKE TCX+Y=120+X*3,C3
1908 :   Y3=Y3+Y3
1918 : NEXT Y
1928 : X3=X3X+X3X
1938 NEXT X
1948 RETURN
1957 :
1968 PRINT MATRIX FORMAT ON SCREEN
1977 :
2000 SCHOLR
2010 PRINT"/D 3 5 7 9 ";CHR$(189);CHR$(
110); " ";CHR$(195); " "
2020 PRINT:PRINT
2030 PRINT"MET W R Y I P * C/H"
2040 PRINT:PRINT
2050 PRINT" @ A D S J L ; CTR"
2060 PRINT:PRINT
2070 PRINT"MLP 4 6 B B ~ -> 2 "
2080 PRINT:PRINT
2090 PRINT"F1 Z C B H , ESCAPE"
2100 PRINT:PRINT
2110 PRINT"F2 S F H K : = COM"
2120 PRINT:PRINT
2130 PRINT"F3 E T U O - + G "
2140 PRINT:PRINT
2150 PRINT" @ SHF X V N , / R/S"
2160 RETURN
2177 :
2188 INITIALISE MACHINE CODE
2197 :
2200 FOR PTR=00+1 TO 03+14
2210 : READ CODE3
2220 : POKE PTR,CODE3
2230 NEXT PTR
2240 RETURN
2257 :
2268 MACHINE CODE DATA
2277 :
2288 DATA 173 , 255 , 55 , 73 , 255 , 32 ,
112 , 219 , 73 , 285 , 141 , 255 , 55 , 94
2297 :
2308 DISASSEMBLY OF MACHINE CODE
2317 :
2328 : 3000 A0 FF 37 LDA #37FF
2338 : 3003 49 FF EOR #0FF
2348 : 3005 28 78 D8 JSR #0078
2358 : 3008 49 FF EOR #0FF
2368 : 300A 60 FF 37 STA #37FF
2378 : 300D 48 RTS

```

back into the accumulator, the X and Y registers are preserved.

The keyboard matrix is shown in Figure 1. To select a column for reading, that column should be pulled low by writing a zero into the appropriate bit, while all the other columns should be held high, i.e., bit value 1. Therefore to select column 2, the number required is, in binary 01110111 or 01F or decimal 251. If no key on that column has been pressed, then the number returned will be 01110111 or 01F or decimal 251. If, however any key on that column has been depressed, then the appropriate bit(s) will be zero, for example, if both 'C' and 'T' were pressed, then the number would be 00001111 or 04F or decimal 175. To examine every key, requires the routine to be called eight times, once for each column. To detect whether ANY key has been pressed, send a zero to all columns, and if the returned value is not 255 then one, or more, keys must be pressed, although which key(s) will not be obvious.

The advantage of using this routine rather than the BASIC GET and GETKEY statement lies in the ability to detect and use multiple keypresses. PR000-8844-1 serves as a useful example and worked. When RUN the program presents



# Program Listing 2

```

10 REM DIS SOLDIERS
11 REM
12 REM BY GEREK MOODY APRIL 1985
13 REM
200 I
300 DIM PLOC(1,500)
400 TCM=9875
500 SCHEX=0
600 CX=14000
700 GOSUB 2000
800 POKE PLOC,01
900 POKE PEX,07
910 I
920 REM CONTROL ROUTINE
930 I
1000 DO UNTIL T
1100 I GOSUB 200
1200 I GOSUB 500
1300 I GOSUB 300
1400 LOOP
1500 IF T=PIN THEN SC=SC+1000:ELSE SC=SC+1000
1600 GOSUB 2000
1700 END
1800 I
1900 SET MOVES AND UPDATE POSITIONS
2000 I
2100 SCX=0
2200 GOSUB 500
2300 PIN=PIN+MOVEX
2400 SCX=0
2500 GOSUB 500
2600 PEX=PEX+MOVEX
2700 RETURN
2800 I
2900 INSERT RANDOM 'STARS'
3000 I
3100 IF INT(RND(1)*100) THEN RETURN
3200 STN=TCM+(INT(RND(1)*10000))
3300 IF PEX<STX<132 THEN RETURN
3400 POKE STX,42
3500 RETURN
3600 I
3700 CALL KEYBOARD ROUTINE AND
    CALCULATE MOVE 499 I
3800 POKE CX,SCX
3900 SCX=1000
4000 RX=PEX-1000
4100 MOVEX=0
4200 IF <RX AND 0 THEN MOVEX=MOVEX-40
4300 IF <RX AND 40 THEN MOVEX=MOVEX-1
4400 IF <RX AND 100 THEN MOVEX=MOVEX+40
4500 IF <RX AND 320 THEN MOVEX=MOVEX+1
4600 IF MOVEX THEN SOUND INT(SCX/20+1),200,SCX,4
4700 RETURN
4800 I

```

Figure 1 — Keyboard Matrix

		COLUMNS						
		0	1	2	3	4	5	6 7
ROWS	0	DEL	3	5	7	9	↑	← 1
	1	RET	W	R	Y	I	P	* END
	2	£	A	D	G	J	L	; CTR
	3	←	4	6	8	0	↑	→ 2
	4	P1	Z	C	B	M	.	ESC CRCL
	5	P2	S	F	H	K	:	= C
	6	P3	E	T	U	O	-	+ Q
	7	@	END	X	V	N	,	/ END CTR

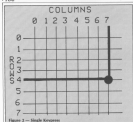


Figure 2 — Single Keyframe



## Program Listing 2 (cont.)

```

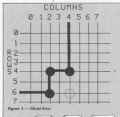
888 UPDATE SCREEN AND LOOK FOR COLLISIONS
889 :
890 TIX=PEEK(PFIX)
891 TZX=PEEK(PZX)
892 IF TIX=44 THEN IF PZX=PLX+8,SIX THEN T=PIX RETURN
893 IF TZX=44 THEN IF MOVX THEN T=PXH RETURN
894 GOSUB 788
895 PLX+8,SIX)=PIX
896 PLX+1,SZX)=PXH
897 POKE PIX,81
898 POKE PZX,87
899 POKE PLX+8,SIX+32
900 POKE PLX+1,SZX+32
901 RETURN
902 :
903 HANDLE ARRAY AND TAIL
904 :
905 SIX=SIX+1:IF SIX=500 THEN SIX=0
906 SZX=SZX+1:IF SZX=500 THEN SZX=0
907 IF TIX=48 THEN EIX=EIX+1:IF EIX=500 THEN EIX=0
908 IF TZX=48 THEN EZX=EZX+1:IF EZX=500 THEN EZX=0
909 IF TIX=40 THEN SOUND 1,500,10:SI=SI+10
910 IF TZX=42 THEN SOUND 2,300,10:SZ=SZ+10
911 RETURN
912 :
913 SETUP SCREEN AND INITIALISE PLAYERS
914 :
915 FOR N=0 TO 29
916 : POKE TCX+N,100
917 : POKE TCX+300+N,100
918 NEXT N
919 FOR N=30 TO 329 STEP 40
920 : POKE TCX+N,100
921 : POKE TCX+N+1,100
922 NEXT N
923 PIX=TCX+300
924 PZX=TCX+310
925 SIX=1
926 SZX=1
927 EIX=0
928 EZX=
929 T=0
930 PLX+8,SIX)=PIX
931 PLX+1,SZX)=PXH
932 GOSUB 888
933 SI=0
934 SZ=0
935 VOL=
936 RETURN
937 :
938 PRINT SCORES AND END GAME
939 :
940 PUDEF 40
941 PRINTOVR(10)
942 PRINT

```

an overview picture of the keyboard matrix, if any keys are held down, then the relevant matrix position is indicated, note that in this case the keys must be held down as the BASIC routine takes a little over a second to record the current position. As the programme is written, it leaves the operating system keyboard untouched, when you have got the programme running properly, add the following line.

78 POKE 181,80:POKE 182,80  
 Save this version of the programme before you RUN it, this stops the vector at 180, and points it to the end of the Interrupt routine, thus bypassing the normal keyboard, and leaving BASIC with no way of reacting to the keyboard. The raw version of the programme does not recognise the RUN STOP key, and allows us to experiment at will.

In PROGRAM 1, TCX points to the top left corner of the matrix as it is printed on the screen. DX, points to the data byte through which parameters are passed to and from the machine code routine, the machine code itself starts at 100+0, SIX is the value that is passed to the machine code routine, SZ is the value returned, SI is a value generated for comparison with



```

2000 PRINTCHR$(166); "          *****"
2010 PRINTCHR$(166); "          "
2020 PRINTCHR$(166); "          "
2030 PRINTCHR$(166); "          "
2040 PRINT "          "
2050 PRINTCHR$(166); "          "
2060 PRINT "          "
2070 PRINTCHR$(166); "          "
2080 PRINTCHR$(166); "          "
2090 PRINTCHR$(166); "          "
2100 PRINTCHR$(166); "          "
2110 PRINTCHR$(166); "          "
2120 FOR K=8 TO 100
2130 I GET AS
2140 SOUND 3,30,1
2150 NEXT K
2160 RETURN
2497 :
2498 TITLE SCREEN
2499 :
2500 SCHOL
2510 FOR K=8 TO 15
2520 I PRINTSPC(442);
2530 I PRINT"*****"
2540 NEXT K
2550 PRINT
2560 PRINT"      PLAYER 1          "
2570 PRINT
2580 PRINT"      W          "
2590 PRINT"      Z          "
2600 PRINT"      A          "
2610 PRINT"      S          "
2620 PRINT
2630 PRINT"      PRESS A KEY!!"
2640 PRINTCHR$(19)
2650 FOR K=8 TO 15
2660 I PRINT
2670 NEXT K
2680 PRINT" DOBBLE THE STARS"
2690 PRINT" AVOID THE REST"
2700 DO WHILE AS=""
2710 I GET AS
2720 LOOP
2730 SCHOL
2740 SOUND 1000
2750 RETURN
2997 :
2998 INITIALISE MACHINE CODE
2999 :
3000 FOR PTR=0X1 TO 0X14
3010 I READ CODEX
3020 I POKE PTR,CODEX
3030 NEXT PTR
3040 RETURN
3497 :
3498 MACHINE CODE DATA
3499 :
3500 DATA 178 , 255 , 55 , 73 , 255 , 32 , 112 , 218
      , 73 , 255 , 141 , 255 , 55 , 55

```



FX, it is used to discover the state of each bit of FX, CS is the character coded in the screen display, it is set to either a space, or an inverse space according to requirements. You will notice that two of the arrow characters, which are not included in the standard character set, had to be connected from others. If you inspect the disassembly of the machine-code routine, you will notice that the parameters are passed in an inverted form, i.e., every 1 has been replaced by a 0, and vice versa, this was done in order to simplify the BASIC logic, the values passed to the keyscan routine are correct.

The keyboard is immense—a set of switches that permit connections between eight lines representing the columns, and eight that represent the rows. By applying a voltage to one of the columns, and searching for a corresponding voltage on the rows, then a closed switch can be detected; for example, if the space bar is closed, the voltage applied to column 7 can be detected on row 4, see FIG 1. So far so good, but we have a problem, when you run PROGRAM 1 you may have noticed that sometimes keys were indicated as having been pressed when you had not touched them, why? Look at FIG 1, on a simple matrix like this it is possible for switches to mask each other, if 'C', 'T', and 'M' are pressed together then 'O' is detected as well, go on, try it! What happens in this, when a voltage is applied to column 4, the closed 'M' switch passes it onto row 4, then the closed 'C' passes it onto column 1 and finally the closed 'T' passes it onto row 6, where it is detected by the keyscan, of course a voltage applied to column 4 and detected on row 6 means that 'O' has been pressed, doesn't it! This phenomenon will not often cause problems, but it has to be borne in mind when selecting control keys for games etc., it would not do, for instance, if hyperspace were selected each time up, left, and the fire keys were pressed simultaneously. As a tip off, next time you run an unprotected BASIC programme, press 'N', 'M', and the space bar all together, hey presto, RUN STOP!

PLCarray)	List of player addresses.
TCOL	Top left corner of screen.
OP	Parameter address (as programme 1).
IP1	Player 1 position.
IP2	Player 2 position.
S1	Player 1 score.
S2	Player 2 score.
T	Flag, 0, or 'trade position'.
MP	Parameter passed to machine code.
MOV15	Player position update value.
ST1	Random position of new view.
R1	Value returned from machine code.
TT1	Contents of new player 1 position.
TT2	Contents of new player 2 position.
ST1	Player 1 list start pointer.
ST2	Player 2 list start pointer.
ET1	Player 1 list end pointer.
ET2	Player 2 list end pointer.

Note that PLCarray) is implemented as a circular list.



Let's have a practical example. PROGRAM 2 is a two player game that requires the detection of eight keys, several of which might be in use at any time. Each player has to have controls for up, down, left, and right, and in addition we must permit diagonal movement. To simplify the input routine, it is desirable that all of one player's controls should be on one column. It so happens that if we use column 1 for player 1, and column 5 for player 2, then there are suitably placed keys for both players on rows 1, 2, 4, and 5. The machine code in PROGRAM 2 is identical to that in programme 1, and the keyboard is made in the subroutine at line 508. This time, rather than looking at the whole keyboard by means of eight column scans, only two scans are made, and the rest of the keyboard is ignored, however the integral vectors are untouched, ROM STOP will work.

The programme is structured for clarity rather than speed, but despite this the game is quite playable.

# The WINNERS

of the ASP DREAM HOLIDAY Competition

Argus Specialist Publications Ltd. are pleased to announce the winners of the fabulous Dream Holiday Competition.

(As seen in the competition entry)



## First Prize

A holiday anywhere in the world up to a value of £2,500 (tax-free) awarded to Mr K. Gougherty of 7 Riverside Road, Bideford, Devon, Devon PL6 2JL.

## Second Prize

A very latest portable video camera recorder (worth over £800) given to Mr C. E. Duffy of 104-106 High Place, Boreham, Essex SS6 7DS.



## Third Prize

The most popular BBC Model B Micro computer plus software package, awarded to Martin P. W. Dwyer of 11 Ladbroke, Bath, Somerset BA2 2JL.

## Fourth prize

A superb Minsley 500 camera with 35mm lens and flashgun, Mr Lee, Salford of 1, Alderley Road, Viper Village, Macclesfield, Cheshire SK10 2ST.

ASP would like to thank everyone who entered the competition, and CONGRATULATE Mr Gougherty for his winning entrance which was a prize-winner.

"...to combat boredom by the beach, keep magazines in easy reach!"

# SPRITE IDEAS

When you are designing a game one of the longest jobs is designing the sprites. If you are good at art then fine, if not your next monster will probably end up looking like a square box with legs.

Now, Your Commodore comes to the rescue once again with Sprite Ideas. If you have designed any sprites for games and you don't mind other people seeing your masterworks then why not send them into us. Each month we will be offering \$10 for the best entries.

Your sprites can be anything at all (within reason), if you've designed a series of animated characters then send in the lot. We'd love to have a look at them.

So, next time you are after an Oge to put in your new game, have a look in this section of the magazine and you may find just what you are looking for.

WOLVES

LEE GOODMAN  
BIRMINGHAM

```

BWT4,240,0,0,24,0,0,0
BWT6,0,54,0,0,114,0,0
BWT220,0,0,220,0,14,220,0
BWT6,220,0,30,19,0,19,220
BWT6,30,19,0,19,220,0,0
BWT234,0,0,234,0,0,230,0
BWT6,234,0,0,230,0,1,230
BWT6,0,230,0,0,1,230,192
    
```



WOLVES - HOAD

STUART JAMES  
BIRMINGHAM

```

BWT6,10,120,0,0,140,0,0
BWT140,0,0,170,140,0,0,140,0
BWT20,124,0,60,124,12,0,114
BWT6,0,90,60,170,170,60,10
BWT140,60,10,174,0,10,174,0
BWT6,170,200,120,174,10,120,174
BWT6,120,200,0,10,160,0,0
BWT140,0,1,170,0,0,1,160
    
```



WOLVES L000

STUART JAMES  
BIRMINGHAM

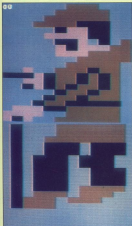
```

BWT6,10,160,0,0,160,0,170
BWT160,0,170,160,10,170,160,0
BWT200,240,60,240,240,10,240,220
BWT6,0,240,220,0,220,220,0,220
BWT60,0,60,60,0,170,0,0
BWT160,160,10,160,240,10,170,0
BWT6,0,0,0,0,0,0,0
BWT6,0,0,0,0,0,0,0
    
```

WATSON-HAGS

DATA0,10,128,0,40,160,0,170  
DATA040,0,171,240,0,20,240,0  
DATA080,240,0,40,96,0,28,70  
DATA120,170,0,2,170,0,0  
DATA160,0,41,200,200,200,194,0  
DATA200,184,10,200,184,0,10,200  
DATA240,200,180,0,170,200,10,200  
DATA280,0,10,180,110,41,110

STUART JAMES  
BETHENHART



WATSON - LEE

DATA00,40,128,40,42,160,40,170  
DATA040,40,170,200,40,200,200,0  
DATA080,240,0,1,241,240,0,1,240,200  
DATA120,240,200,0,190,200,0,190  
DATA160,0,162,160,0,100,120,100  
DATA200,0,0,0,0,0,0,0  
DATA240,0,0,0,0,0,0,0  
DATA280,0,0,0,0,0,0,0

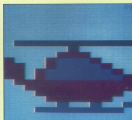
STUART JAMES  
BETHENHART



GHOST

DATA0,14,0,0,204,0,0,170  
DATA40,0,170,0,0,1,128,128  
DATA80,128,127,191,0,42,204,0  
DATA120,0,240,0,0,190,0,1,190  
DATA160,0,200,0,1,204,0,1  
DATA200,0,1,204,0,0,204,0  
DATA240,0,0,1,204,0,1,200  
DATA280,0,200,0,0,200,120

LEE GOODMAN  
BETHENHART



CHOPPER

DATA0,0,40,170,170,0,0,0  
DATA40,10,0,44,127,44,0,127  
DATA80,0,200,244,0,200,200,0  
DATA120,0,0,0,0,0,0,0  
DATA160,0,0,0,0,0,0,0  
DATA200,0,0,0,0,0,0,0  
DATA240,0,0,0,0,0,0,0

D. HERRICK  
CHOPPER

DATA0,0,0,0,0,0,0,0  
DATA40,0,0,0,0,0,0,0  
DATA80,0,0,0,0,0,0,0  
DATA120,0,0,0,0,0,0,0  
DATA160,0,0,0,0,0,0,0  
DATA200,0,0,0,0,0,0,0  
DATA240,0,0,0,0,0,0,0  
DATA280,0,0,0,0,0,0,0

# EASY ENTRY EASY ENTRY

```

100 INPUT "COLS,DOWN,START ADDRESS: ";ADR
110 PRINT "DOWN=PRESS ENTER ALL RIGHTS."
120 PRINT"DOWN=SPACES WILL BE ENTERED AUTOMATICALLY"
    (DOWN)
130 PRINT:PRINT ADR;" ";:GOTO 240
140 IF DR="D" THEN GOTO 130:GOTO 100
150 IF DR="P" THEN GOTO 130:GOTO 100
160 DR=READ DATA AND STORE
170 CHECK=ADR-INT(ADR/256)*256
180 FOR D=1 TO 16 STEP 5
190 DR=ADR+DR*5;D;D;D+DR/100
200 CHECK=(CHECK+DR)*255
210 IF X=255 THEN DR=0
220 POKE ADR,ADR+DR+1:NEXT D
230 PRINT:PRINT(ADR,21)
240 IF DR=CHECK THEN GOTO 200:GOTO 130
250 GOTO 130
260 DR=""FOR D=1 TO 16:FOR L=1 TO 3
270 GET DR;IF DR=""GOTO 270
280 IF DR="P" THEN D=L+1
290 IF DR="P" THEN D=L+1
300 IF DR="P" THEN D=L+1
310 DR=DR+5
320 PRINT DR:PRINT L:PRINT " ";:NEXT D:GOTO 130
330 DR=0:SAVE YOUR FILE NOW
340 PRINT"COLS,DOWN,RIGHT=SAVE FILE:DOWN=1"
350 INPUT:FILE NAME: ";FN
360 IF FN="" OR LEN(FN)<8 OR FN="0" THEN RETURN
360 INPUT"COLS,DOWN,RIGHT=OFF:AP: DR=PC,DOWN=DOWN:OFF:
    (K = LEFT)";DR
370 D=0:IF DR="" THEN D=0
380 INPUT"DOWN=START ADDRESS IN DECIMAL: ";D
390 INPUT"DOWN=END ADDRESS IN DECIMAL:PC(L)";E
400 TO=DR+TO+(D-255)*256+PC(L)-LEN(TO)
    (POKE TO,TO)
410 POKE TO,TO+HEX(702)*256+POKE TO,LEN(TO)
420 D=0:GOTO
430 POKE TO,1:POKE TO,0:POKE TO,0:D=0:GOTO 4044
440 POKE 254,0:POKE 255,0-POKE 254,1:POKE TO,255
450 POKE TO,0+0:POKE TO,0+0-POKE TO,254:GOTO
    (D=0:GOTO 4044)
460 DR=0:LOAD DATA NOW
470 INPUT"COLS,DOWN,RIGHT=FILE NAME: ";FN
480 IF FN="" OR LEN(FN)<8 THEN RETURN
490 INPUT"COLS,DOWN,RIGHT=OFF:AP: DR=PC,DOWN=DOWN:OFF:
    (K = LEFT)";DR
500 IF DR="" THEN D=0:GOTO 4044
510 D=0:IF DR="" THEN D=0
520 LOAD FILE,1:RETURN
530 PRINT:PRINT"COLS,RIGHT=DOWN:ADR=ADR+12
540 POKE 2476,15:POKE 2477,10
550 POKE 2478,100:POKE 2479,45:POKE 2480,00
    (POKE 2481,17
560 FOR D=1 TO 255:NEXT
570 POKE 2476,0:POKE 2477,0:POKE 2478,0
580 RETURN

```

We make life easier

for you with our

machine code entry

program.

THE WORST THING ABOUT Machine Code programming is entering thousands of numbers and then finding that the program will not work. There is nothing else that you can do apart from go through all of the listing trying to locate that mistyped character which prevents the program from working correctly.

Now there's an easier way to enter your machine code programs. With the Year Commodore machine code entry program, each line of numbers is checked as soon as you press return. If you have made a mistake you will be asked to re-type the last line. Another added bonus is that you can save what you have entered at any time to tape or disk and carry on where you left off next time you come to your computer.

## Using the Loader

Before you type in any machine code program you must have typed in the machine code entry program and have it saved onto tape or disk. When you want to enter any of the machine code programs that

are printed out in the form used by this program you must LOAD it into your computer. When you RUN the program you will be asked for the start address of the program. The start address is the first number in any machine code listing that appears before the colon (e.g. 49152). You simply type in this number and press return.

All that you have to do from then on is type in all the numbers on a line. Do not type any spaces and do not type return, the program will do all of that for you. If you have made a mistake on any line the computer will ask you to type the line again. Once the line is entered correctly the computer will automatically prompt you for the next line of data.

## Saving and Loading

You can save your data to tape or disk at any time by simply entering the F1 key as the first character on any line. You will then be asked for the start and end address of the save. The start address is the first number in the listing as already mentioned. The end address is the number of the last line plus 17. Don't forget to add 17 as the last entered will not be saved.

To load back a program that you have saved you simply have to enter the F1 key as the first item on a line. You will then be asked for the name of the program.



1. 4444444444 2. 4444444444 3. 4444444444 4. 4444444444 5. 4444444444 6. 4444444444 7. 4444444444 8. 4444444444 9. 4444444444 10. 4444444444

Dave Crisp takes a look at a handy utility that will allow you to re-align your disk drive.

## How's Your Disk Drive?

ASIDE FROM THE FACT THAT IT'S slow, does it also fail to load some of your software?

Well, the problem could be head alignment. That is, the part that moves over the disk in your drive. Because of the two types of metal used in the head mechanism high temperatures cause the metals to expand at different rates. The result of this is a floppy head. The answer to this is to let the drive cool down and hopefully things will be OK again.

The other cause of mis-alignment is more serious. Have you noticed with some protected software the drive makes a hammering noise like a machine gun? This noise is the mechanism being 'bumped' against the stop. This repeated hammering will eventually knock the head out of position.

Would other computer owners put up with a situation where software can damage the hardware?

If you are suffering from a badly mis-aligned head then this software from Iyengar Micros may be the answer.

## No Special Equipment

As far as I know this is the first disk of its type. For the first time it is possible to check and remedy head alignment without equipment such as oscilloscopes.

The disk itself has had its tracks recorded 'off-line' so that the program can interpret what it reads into a measurement.

When I received the copy of the software I was relieved as my heads were so mis-aligned that it was getting to a point where I was finding it difficult to get a directory of a disk let alone run programs with any degree of confidence.

## The Tests

There are two main tests:

### 1 Speed and Clamping test

The speed at which the disk rotates in the drive is very important and the T&M exam shows quickly whether this could be the first of your problems.

The software takes 10 measurements of your drives speed. This is then converted into an average speed. This average speed should be within  $\pm 3$  to  $\pm 5$  of



300 rpm (15%). If this is OK the difference between the fastest sample and the slowest sample is noted and if this is greater than 0.6 then this would result in a failure.

### 2 Radial Head Alignment

The program reads what data it can from off-track disk and shows the result on a chart.

The chart is made up of a display of asterisks which shows at a glance how far out your drive head is. The chart will also show whether the mis-alignment is to the inside or outside of the track.

In the manual provided there are 34 read-outs showing results and a remedy to whatever result you get you should be able to find a chart which looks something like the one you obtain.

There is a third test which checks the position of the TRACK 1 STOP.

This is a metal switch which stops the head moving further back than track one. It is possible for this casting to become misplaced which obviously results in the head mis-aligning with that track.

The program does a stop check and shows on the chart whether you need to re-align the stop.

A printout of a chart is shown below.

## Doing The Work

After doing the test you decide that some work on your drive is required this

manual will take you through the procedure step by step. There is not enough space here to allow me to go through the procedure that you would need confidence in your ability to poke around with a screwdriver without damaging anything. If in doubt find somebody who feels a little more confident.

## Silence The Gun

You will also find in the manual a small envelope containing two sets metal springs. These replace the standard head stop. This is a simple job and one that is worth doing.

This does not stop the 'hammering' of the drive but it does make the hammering very quiet and less damaging. After fitting the 'quiet stop' run the test again to ensure that it is in the correct position.

## Conclusion

This is a good thing if you envisage problems. Of course if your head is so far out of line already you will not be able to load the diagnostics anyway. It is a useful thing to have and one which most Commodore owners would find useful at some time in their drive's life.

At £28.95 it is not cheap but could pay for itself. It is available from: Iyengar Micros, Telephone 0286 41069 or 021-438344.

Listings will be much easier to enter with our new system.

COMMODORE LISTINGS ARE RATHER well known for the horrible little black blobs that always abound. Unfortunately the graphics characters which are used to represent graphic and control characters do not reproduce very well and they are also difficult to deal on the Commodore keyboard.

In future all control and graphics commands will be replaced by a mnemonic within square brackets. This mnemonic is not typed out as printed in the magazine but rather the corresponding key or keys on the keyboard are pressed. For example [RIGHT] means press the cursor right key, you do not type in [RIGHT]. All of the keywords, what keys to press and how they are shown on the screen are shown below.

Any character that is accessed by pressing shift and a letter will be printed as (letter).

[SA] shift and A

[S-] shift and -

Any character that is accessed by pressing the Commodore key and a letter will be printed as (letter).

[CA] Commodore and A

[C-] Commodore and -

[C+] Commodore and +

# LISTINGS

If any characters are repeated the mnemonic will be followed by a number. This number is how many times you should enter the character. Any number of spaces over one will also be represented in this form.

[RIGHT10] press cursor right 10 times

[C+10] press Commodore and + 10 times

[SFC10] Press the space bar 10 times

Any other characters should be easily recognisable for example CTRL-PA means press CTRL and P and LEFT-ARROW means press the left arrow.

Any number of mnemonics can be enclosed in brackets for example

[SA10,SPCR10]

means type 10 shift A's 10 spaces and another 10 shift A's.

Mnemonic	Symbol	what to press
[RIGHT]		left/right
[LEFT]		shift left/right
[UP]		Shift & up /down
[DOWN]		up/down
[F1]		F1
[F2]		shift & F1
[F3]		F3
[F4]		shift & F3

Mnemonic	Symbol	what to press
[R]		R
[N]		shift & R
[7]		7
[8]		shift & 7
[CLEAR]		shift & CLR /HOME
[HOME]		CLR/HOME
[F5ON]		CTRL & F
[F5OFF]		CTRL & F

Mnemonic	Symbol	what to press
[BLACK]		CTRL & B
[WHITE]		CTRL & Z
[RED]		CTRL & D
[CLR]		CTRL & C
[PURPLE]		CTRL & S
[GREEN]		CTRL & G
[BLUE]		CTRL & F
[YELLOW]		CTRL & E





## ACCESSORIES

Quadrant III (negative)	08.7%
Quadrant II (negative)	08.7%
C-10 subgroup (Not shown)	03.8%
Fixed (arch) (INC. STL)	
C-10 STL	03.8%

Capex: \$PI Digital interface  
Capex all T1 lines, 10/1000, etc.  
IC controlled. No software  
needed. Just type text. If it  
loads, CPU will copy it. Requires  
several databases. 12 months

## CATALINE

**FOR HIRE**

**BUY OR SELL** your unwanted software, books, and hardware. We put you directly in contact with each other in your area. Details: Second Hand Computer Sales Agency, 38, Bowdoin Alley, Waltham, MA 02154. (617) 852-0000. [www.shcsa.com](http://www.shcsa.com)

## SOFTWARE APPLICATIONS

## HOME ACCOUNTS

Classroom Software  
Prospect,  
Beaville, Appalachia  
November 1995 800  
Tel. 800-441-0120

## FOIA b 7 - D

**THINK TIGHTER! WE'VE GOT YOU!**  
Only 11 days of \$2.99! Tomorrow, Thursday  
5-11pm. Saturday 12-10pm. Sunday 12-6pm.  
\$2.99! (excludes 5-10pm, Happy Hour  
12pm-2:30pm, 4-6pm) to  
Bottles Unlimited, 875 New St.  
at Highland St., Burlington, VT 05401  
Tel. (802) 255-4444  
Call today to reserve.

**ZAP THE COMPETITION!**  
**PZZT!**  
To Advertise: (800) 614-5200

**DON'T LEAVE IT TO CHANCE!  
GIVE YOUR BUSINESS A BOOST  
BY ADVERTISING IN  
YOUR COMMODORE  
PHONE STUART TAYLOR ON 01-437 0699  
FOR DETAILS**

**Test results released** .....

Please use the following information to answer the question:

A NEW CLASSIC

FROM PAUL WOAKES  
AUTHOR OF ENCOUNTERS

# YOU CHOOSE THE ACTION IN MERCENARY

Mercenary — a unique combination of flight simulation, adventure and arcade fun. You choose the action.

There is complete freedom of movement in a truly three-dimensional vector-graphic environment. Graphics of exceptional speed create a very realistic experience.

You'll never play the same game twice. Random elements attempt to seal your fate. Your interaction is crucial.

Mercenary presents an absorbing challenge that you will accept again and again.

## MERCENARY ESCAPE FROM TARG

Fly above the planet. Tap in a series of code, break out the multitude of adventures for the planet surface and your quest for essential weapons. Search for information, undertake complex missions at your leisure.

Become involved in the continuing conflict between the Targans and Neutronics. Accept your part assignments or work independently to gain credit.

Up only have one life blood — the ever-reliable Radar, an artificial human PC, who guides you in your travels.

Runse your abilities as you will. To succeed you have to outguess the Targ.

INITIAL RELEASE  
FOR COMMODORE 64 ATARI 438K 800 XL 130 HE  
CASSETTE £9.95 DISK £12.95

POUNCEY SOFTWARE LTD. SALES 142 ALCESTER ROAD BIRMINGHAM B15 2JG

**NOVAGEN**

# BUSICALC 3

**- the sophisticated spreadsheet !**

Easy to learn, easy to use - something that can't be said of many business programs. But it's true of all the programs in the BUSICALC series.

BUSICALC 3 can handle all sorts of jobs - budgets, expenditure analysis, stock lists, price lists, and product costing are just a few of the possibilities. Three-dimensional formulae automatically access data stored on disk, so that you can easily pull together information from several different sheets, and summarise or manipulate it.

It's simple to transfer data to other programs, such as Easy Script. And you can use virtually any printer with BUSICALC 3, whether dot matrix or daisy wheel, Commodore or non-Commodore.

For the CBM 64 and PET/CBM 4000 & 8000 series.

Available through dealers or from:  
**Supernoft, Winchester House, Canning Road, Harrow HA3 7LJ**

**Phone 01-861 1166 for more details and a free catalogue.**

